Rochester Institute of Technology

RIT Scholar Works

Theses

11-1-1996

Image quality metrics and optimum focus criteria for visual optical systems

Xiaoxue Cheng

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

Cheng, Xiaoxue, "Image quality metrics and optimum focus criteria for visual optical systems" (1996). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

IMAGE QUALITY METRICS AND OPTIMUM FOCUS CRITERIA FOR VISUAL OPTICAL SYSTEMS

by

Xiaoxue Cheng

M.S. Sichuan University, China (1988)
B.S. Sichuan University, China (1985)

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Center for Imaging Science
in the College of Science
at the Rochester Institute of Technology

November 1996

Signature of the Author ______________________________

Accepted by Harry E. Rhody 12/5/96
Coordinator, M.S. Degree Program Date

CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NY

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Xiaoxue Cheng
has been examined and approved by the thesis committee
as satisfactory for the
Master of Science degree

____
Dr. Pantazis Mouroulis, Thesis Advisor

____
Dr. Jeff Pelz

____
Dr. Zoran Ninkov

11-12-96
Date

THESIS RELEASE PERMISSION

Rochester Institute of Technology
College of Science
Center for Imaging Science

Title of Thesis: Image Quality Metrics and Optimum Focus Criteria for Visual Optical Systems

I, Xiaoxue Cheng, hereby grant permission to the Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature ________________________________

Date Dec. 4, 1996

IMAGE QUALITY METRICS AND OPTIMUM FOCUS CRITERIA FOR VISUAL OPTICAL SYSTEMS

by
Xiaoxue Cheng

Submitted to the
Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science degree
at the Rochester Institute of Technology

ABSTRACT

Image quality metrics for visual instruments were examined in terms of their through focus behavior in the presence of various aberrations, and their correlations with available subjective performance data. The contrast sensitivity measurements were performed using rotationally symmetric, variable contrast difference-of-gaussians (DOG) targets, viewed through specially designed telescopes that presented various amounts of monochromatic aberrations. Then the contrast sensitivity ratios were correlated with the image quality metrics of the telescopes. The results show that an appropriately defined integral of the instrument-observer MTF (called MTFa) correlates well with subjective performance in most cases and predicts the optimum focus best; the radius that encircles 84% of the energy of the point spread function (called R_{84}) gives good correlation in some cases including the DOG experiment.

ACKNOWLEDGMENTS

I am most grateful to Dr. Pantazis Mouroulis for introducing me to the field of visual image quality. His knowledge and guidance made this thesis possible.

My grateful thanks are due to Kevin Lyons for the design, testing and calibration of the telescopes, as well as participating in the psychophysical experiment.

My grateful thanks are also due to John Handley for his much help with the FFT program, Taek Kim for much of the software for controlling the monitor, and Guoheng Zhao for the monitor calibration.

DEDICATION

This thesis is dedicated to my dear husband and son.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

By visual optical systems we mean optical systems which present an image to the eye with no intermediate screen involved, such as telescopes, periscopes, binoculars, sighting instruments, magnifiers, etc. Measuring the performance of such visual instruments is a complex problem. The complexity is caused by three main factors: first, the coherent coupling between instrument and the observer's eye. The coherent coupling means that the optical elements of the eye effectively form an integral part of the instrument under test, thus any test method must take into account not only the instrument itself, but also the eye and the interaction between instrument and eye. Second, the eye is a dynamic system that constantly changes its optical properties such as pupil size and focal length etc. to adjust to different visual tasks. Third, the eye is the intermediate stage of the entire imaging process between the instrument and the observer's brain; the neural processes that occur after the retina can affect the perceived image quality. Therefore, the perceived image quality cannot be directly measured by an external machine; the assessment has been forced to rely on the observer's perception. In fact, the quality assurance of most visual optical instruments in current production depends heavily on the ability and experience of human inspectors using test charts. However, neither the test charts nor the inspectors used by different manufacturers are standardized or inter-related. It is thus almost impossible to relate the performance of one visual instrument to another. Such a disadvantageous state has led many to attempt to establish links between visual performance and some common instrumental metrics[1,4-9] in the hope to find an objective metric which can characterize the image quality of a visual instrument. If such a metric were available, it could be used as part of the merit function during design optimization, and it could also serve as a means of testing and qualifying an instrument in a way that approximates the expected observer performance.

Past attempts to define such metrics have ended in controversy. About two decades ago, the British Ministry of Defence (MOD) selected the modulation transfer function (MTF) measured at a single spatial frequency[1-3], without the supporting published data linking MTF to visual performance. In the recent literature, Giles[4] investigated the effects of approximately one and two waves instrumental spherical aberration, coma, and astigmatism on the total system (instrument-plus-eye) MTF using three bar resolution targets and square-wave gratings, and found good correlation between subjective performance (SP) and the MTF, measured at the plane of minimum wavefront aberration variance. Subsequently, Burton and Haig[5] found no correlation between SP (subjective resolution tests) and the MTF using high contrast Cobb charts, but the types and amounts of aberration were not stated in that study. Mouroulis[6] used sinusoidal gratings and three bar targets and found some correlation between SP and the MTF, but warned that no correlation should be expected if the accommodative response cannot be predicted. Later Haig and Burton[7] repeated their claim of no correlation between MTF and SP, but suggested the Strehl ratio as an alternative at least for systems with Strehl ratio greater than 0.8, using a picture of a tank and a pseudorandom pattern as the targets. Recently, Haig and Williams[18] emphasized again that the Strehl ratio is a good metric to use when it is greater than 0.8 which actually restricted the applicability of Strehl ratio only to instruments with very high image quality. On the other hand, Mouroulis and Zhang[8] proposed two new image quality metrics that showed better correlation with SP than the Strehl ratio over an extended range of image quality, while at the same time disqualifying the wavefront variance as a useful metric. The aberration conditions Mouroulis and Zhang examined were coma, astigmatism and their combinations which gave Strehl ratio in the range of 0.8 to 0.17. The targets were sinusoids, a narrow bar and an edge. The two new metrics were: (1) the integral of the MTF over a frequency range starting at around 5 cycle/deg, and ending near the highest frequency of interest, but no higher than 20 to 25 cycle/deg, called MTFa; (2) the radius that encircles 84% of the energy of the point-spread function (PSF), called R_{84}.

One of the aims for this thesis is to determine whether the above conflicting conclusions about the preferred metric can be reconciled. We do so by examining the correlations between the various objective image quality metrics and the subjective performance data of the above-mentioned investigations.

Second, irrespective of the choice of metric, we still need to know at what focal plane to do the testing (of visual instrument). This important fact has not received the careful attention it deserves. Clearly, the testing plane should coincide with the plane that the observer chooses through accommodation. Therefore, the research should establish what determines the accommodative response in the presence of various aberrations. For example, in objective MTF testing the focus is often chosen by maximizing the response at a single spatial frequency.[5] But is there any evidence that the eye adopts a similar accommodative strategy? We investigate this problem by examining the through-focus behavior of alternative image quality criteria which are Strehl ratio, SQF (Subjective Quality Factor), the RMS (root mean square) of wavefront aberration variance, MTFa and R_{84} in the presence of various aberrations.

Another problem with all psychophysical experiments conducted in search of a suitable objective metric is that the target and visual task used may exert considerable influence on the results. So far, the main targets used are 'line" targets such as bar and grating[4,6,8] which present primarily one dimensional detail. As we shall see in chapter 4, the correlation with the MTFa and R_{84} was obtained using mostly 'line" targets, whereas the results of Burton and Haig[7] in favor of the Strehl ratio were obtained using targets with two-dimensional detail. Thus it is desirable to test all metrics against two dimensional targets, in order to ascertain whether the different conclusions are due to the use of different targets.

So what kind of two dimensional target should we use in the experiments to fully evaluate the performance of visual instruments? We know as a result of the nonlinearity of the eye, that no single target and task suffice to characterize fully the observer/instrument system performance. A relatively complete description may be obtained if one employs all of the following tests: 1) a contrast sensitivity test, typically using low-contrast sinusoids at several frequencies, 2) a high-contrast resolution test, using a three bar or other resolution target, and 3) a discrimination test using more complex targets, perhaps suited to a particular application. The predominance of directional aberrations off-axis (e.g. astigmatism or coma) further complicates matters, necessitating that tests be conducted at several azimuths for directional targets such as sinusoids.

It is therefore of interest to investigate the extent to which alternative targets and tasks can reduce the testing time and effort, without seriously compromising the validity of the results, especially for the contrast sensitivity test which is the most time consuming of the three tests mentioned previously. It is with all these factors in mind that we identified a circularly symmetric difference-of-gaussians (DOG) target as a candidate. The use of DOG in vision research has been discussed by Regan[20]. Its advantages from our viewpoint are that it has a broad but controllable spatial frequency spectrum and the lack of preferential orientation. Thus we hope to be able to substitute contrast sensitivity tests requiring sinusoids at several azimuths and frequencies with a single test using the DOG target. More details about the DOG target we used in our experiment with real telescopes and the correlation results are given in chapter 5.

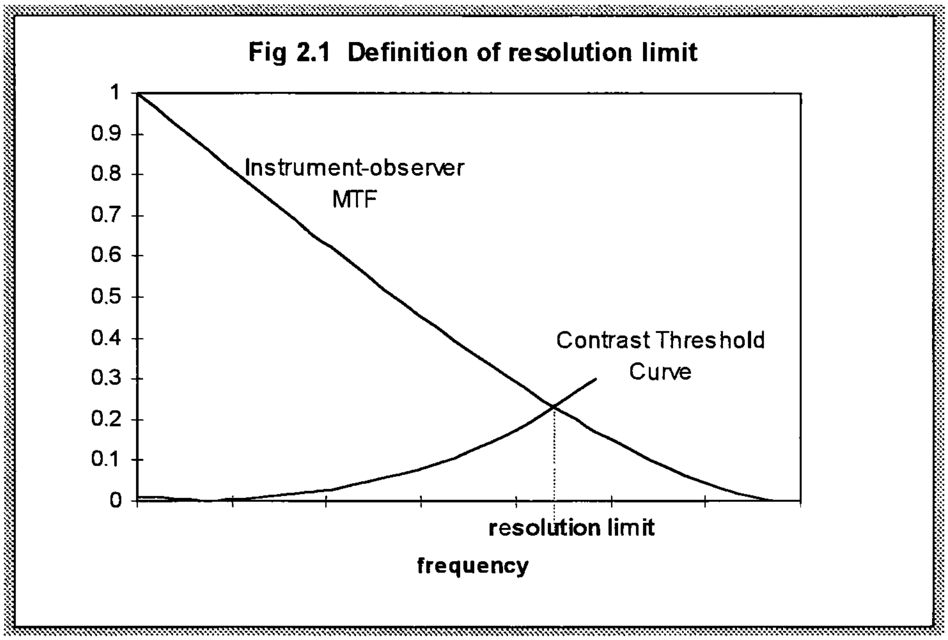
Chapter 2

Possible Image Quality Criteria for Visual Optical Systems

2.1 The MTF and resolution limit

The MTF (Modulation Transfer Function) of a general system is defined as the ratio of the contrast of the output sinusoid grating to that of the input sinusoid grating vs. different spatial frequencies. For optical lens systems, MTF can be calculated as the Fourier transform of point spread function (PSF) or by autocorrelation of the pupil function. The point spread function is the intensity distribution in the image plane of a point object.

The MTF, as a two-dimensional function, differs for different spatial frequencies, field points, and grating orientations and contains a lot of information which must be somehow reduced, if it is to serve as a single figure of merit. We must note the azimuthal variation of the MTF in the presence of non-rotationally symmetric aberrations. The worst MTF can always be taken as an indication of worst-case performance, but for an alternative, any number of conventions can be applied, all of which are essentially arbitrary. In this study we often use the mean of the sagittal, tangential, and 45^0 responses, independent of the type of aberration.



After reducing the azimuthal information, we may further seek to reduce the spatial frequency information. This has been done in two ways: i). take a weighted or non-weighted integral over a specified frequency range, or ii). take the intersection between the MTF that describes the instrument-eye wavefront error, and the contrast threshold or bar detectability curve that describe the fovea-brain system; the intersection of the two curves then represents the limit of resolution of the instrument-observer system, as shown in Fig.2.1. The first way is examined in more details in section 2.4. The second way was proposed by Giles[4], and can be useful if the system specification is in terms of a maximum resolvable frequency.

2.2 The Strehl ratio and rms (or variance) of wavefront aberration

The Strehl ratio is defined as the ratio of the peak intensity value of the aberrated PSF to that of the unaberrated PSF. It is equal to the integral of the transfer function over all frequencies.

If we use W(x, y) to denote the wavefront aberration, then the variance of wavefront aberration, denoted by E, is

$$E = \overline{W^2} - \overline{W}^2$$

where $\overline{W^2} = \frac{1}{A}\iint_A W^2(x,y)dxdy$, and $\overline{W}^2 = [\frac{1}{A}\iint_A W(x,y)dxdy]^2$ And x, y are the exit pupil coordinates, A is the exit pupil area.

For small aberrations, or Strehl ratio $I \geq 0.8$, the variance E and the Strehl ratio I are related by the approximation[23]

$$I = |\ 1 - 2\pi^2 E/\lambda^2\ |^2$$

It is often said that the range of validity of the Strehl ratio is from 0.5 to 1. As far as we can ascertain, the origin of that assertion comes from the work of King[10], who showed that the wavefront variance and the Strehl ratio are not monotonically related if the Strehl ratio is less than about 0.5. But for visual systems, this reasoning is not sufficient, as there is no reason to assume that the wavefront variance is a visually significant metric (recent results indicate that it is not[8]). The variance or the rms of wavefront was included in this study because it is so often quoted and used.

2.3 The SQF[11] (subjective quality factor)

The SQF is a weighted MTF integral, defined as

$$\mathrm{SQF} = K\int_{10}^{40} \tau(f)\,d(\log f)$$

where f is the spatial frequency, τ is the MTF, and K is a normalizing constant obtained by performing the integration with $\tau = 1$. The integration limits represent cycles per mm on the retina, and translate to around 3-12 cycle/deg for a rear nodal distance of 17mm, typical of the unaccommodated human eye. A two-dimensional extension of the SQF definition also exists, but the one-dimensional one was used in line with the reasoning for the reduction of the MTF azimuthal information. The logarithmic weighting causes the low frequencies to contribute more to the total value of the integral.

2.4 The MTFa and MTFv

The MTFa was defined in [8] as the MTF integral in the range 5-24 cycle/deg. The upper limit is not fixed by any fundamental consideration and should be seen as flexible between 20-25 cycle/deg. It was chosen through the following reasoning: i). the off-axis MTF often drops to nearly zero at around 20 cycle/deg even for well-corrected systems. For example, the system MTF approaches or crosses zero at around 20 cycle/deg for instruments off axis with astigmatism W_{22} of 1, 2 and 3λ or coma W_{31} of 1, 2 and 3λ (refer to Fig 2.2, 2.3, where 3mm of limiting pupil diameter is assumed which results in 95 cycle/deg of the cutoff frequency for the system); and ii). the accuracy of the accommodative response tends to level off after 25 cycle/deg.[12] The differences between 20 cycle/deg and 25 cycle/deg for integral upper limit are less than the noise of the subjective measurements. The lower limit is chosen to be close to or just past the peak of the visual contrast sensitivity function which is around 5 cycle/deg. Frequencies lower than 5 cycle/deg show little attenuation even in the presence of aberrations for instruments of reasonable quality which, therefore, carry little useful information about the instrument image quality.

The difference between SQF and MTFa is not only in the overall frequency range but also in the fact that the MTFa weighs all frequencies equally, so the higher frequencies are much more prominent in the MTFa. Thus the MTFa will be expected to be more sensitive to aberrations and vary over a wider range than the SQF.

Fig 2.2 45^0 azimuth MTF for astigmatism of 1λ, 2λ and 3λ with $W_{20}=-W_{22}/2$

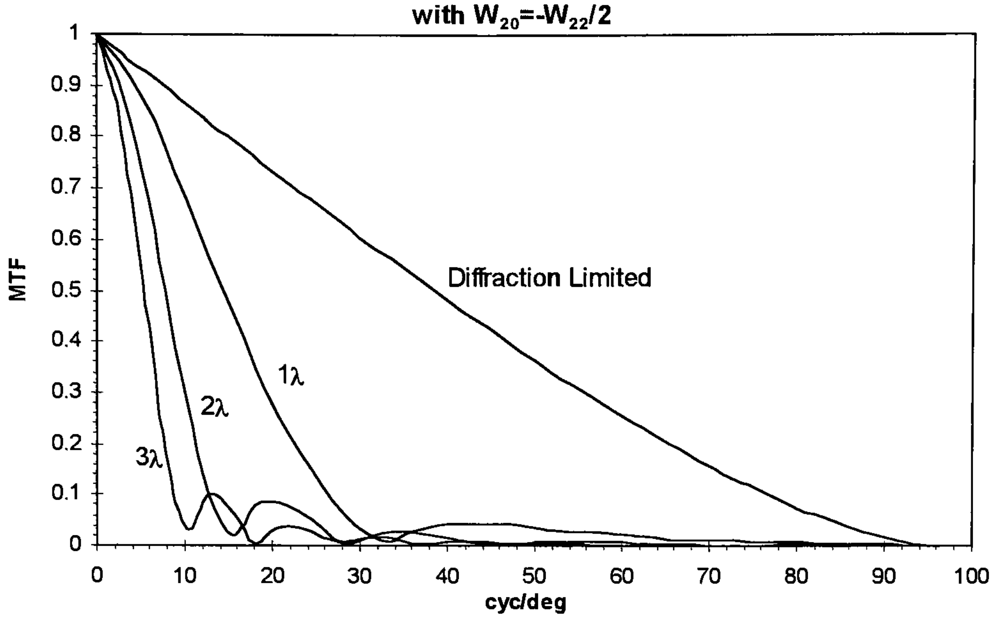
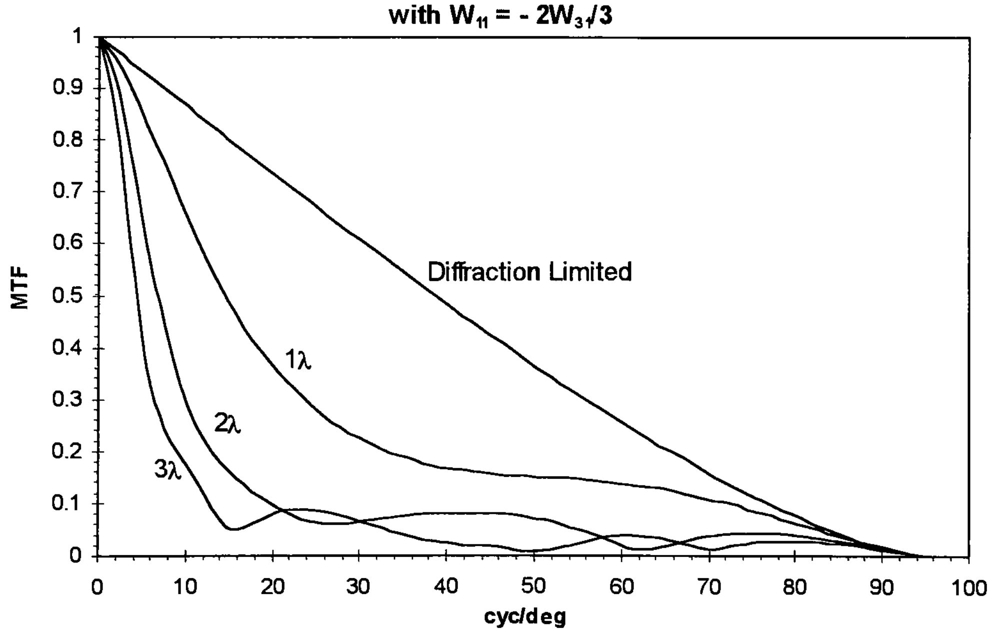


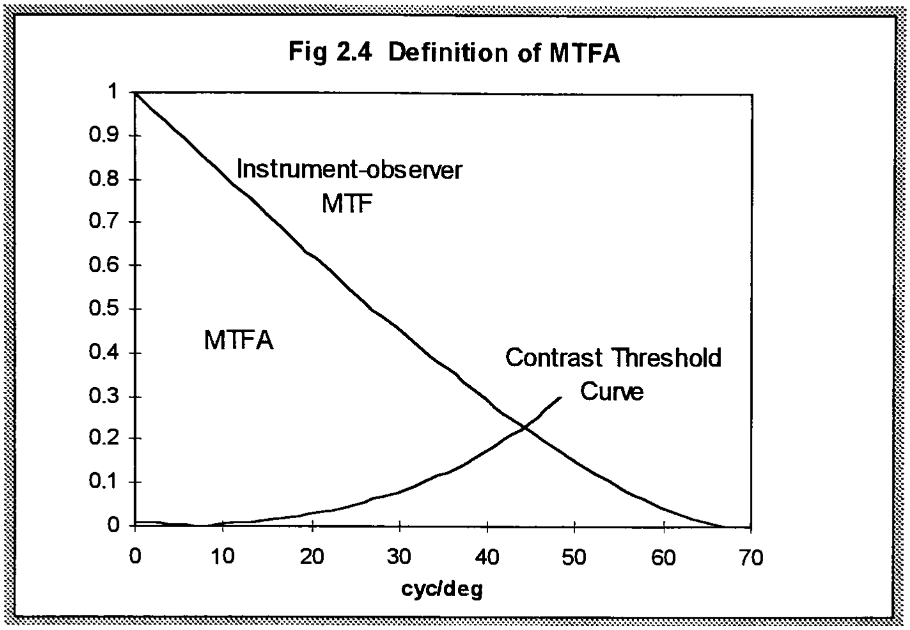
Fig 2.3 Worst case MTF for coma of 1λ, 2λ and 3λ with $W_{11} = -2W_{31}/3$



In the presence of non-rotationally symmetric aberrations, MTF is no longer symmetric in two dimensions. We usually take the worst MTF to calculate MTFa for the worst case, and the mean of four MTFa values at 0, 90, +45 and -45 degree azimuth as an estimate of the overall MTFa in two dimensions. To be more accurate (or less arbitrary), the MTFv concept is introduced here and used in our study later which is defined as the volume under the 2-D MTF surface in the same frequency range as that of MTFa.

2.5 The MTFA

Originally proposed by Charman and Olin[14], developed for photographic systems, MTFA is the area bounded by the instrument-observer MTF and the contrast threshold curve (refer to Fig 2.4). The MTFA has been successful both in aerial photograph interpretation and in raster-scanned imagery. Both cases, however, refer to incoherently coupled systems which have important differences from the coherently coupled systems of our concern. For example, in the case of photographic or display systems, factors such as the noise characteristics of the film/display can raise the contrast threshold. But for coherently coupled systems, no such factors are inherent. As a result, the contrast threshold curve that corresponds to the retina-brain subsystem is not variable, and remains quite low at most frequencies of interest as can been seen in Fig 2.4.



Comparing with MTFa, the high frequency limit of the MTFA, as the intersection between the MTF and the contrast threshold curve, is not always possible to define. This is because, (1) in the presence of non-rationally symmetric aberrations such as coma and astigmatism, MTF will change with azimuth, so does its intersection with contrast threshold curve. For example, for a system suffering from 1.3λ of astigmatism, the points of intersection of the contrast threshold curve with the 45^0 azimuth MTF and the tangential MTF (both at midfocus) are around 34 cyc/deg and 22 cyc/deg respectively, for 2.8 mm pupil diameter. (2) the contrast threshold curve can vary with azimuth by as much as a factor of two[25] Thus even if we have a single MTF over different azimuths, we still have to contend with potentially different contrast threshold curves.

2.6 The radius of a fixed encircled energy

The radius of a fixed encircled energy, R_{84} , was proposed in [8] as a means of characterizing the worst-case performance, even with asymmetric aberrations. The subscript refers to 84% of the encircled energy, which is the energy up to the first zero of the Airy disk. While this looks like a nice convention, there is no *a priori* reason why the radius of 84% or any other value of encircled energy should correlate with subjective performance. In the presence of asymmetric, coma-type aberrations, the center of the circle from which to measure the encircled energy is no longer uniquely defined. One might choose either the peak of the PSF or the centroid. The difference is generally small, except for larger aberrations (e.g. $W_{31} \geq 1.5\lambda$).

Chapter 3

Through-focus Behavior of Image Quality Criteria

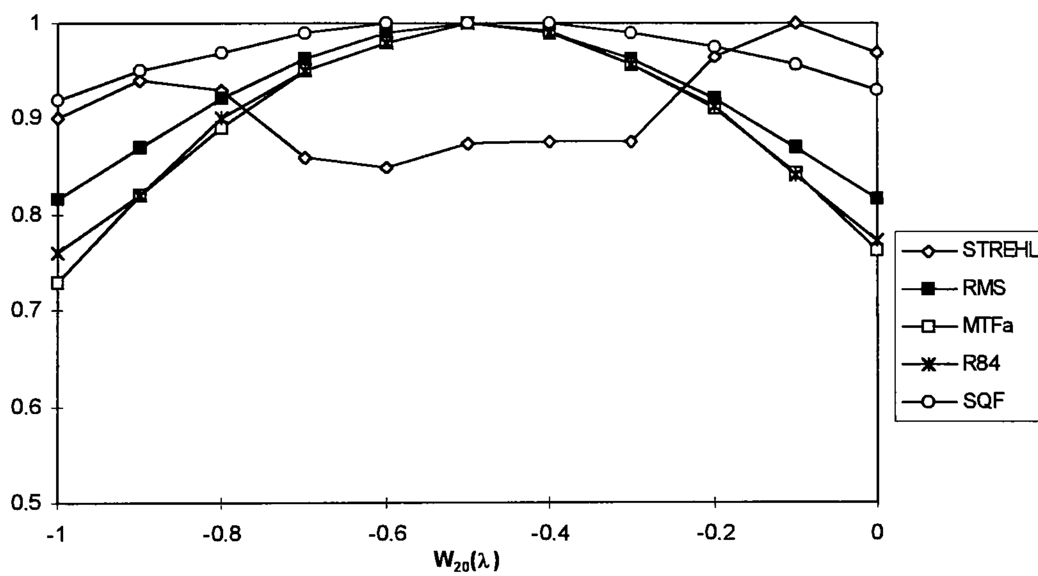
Among the image quality criteria discussed in chapter 2, we will examine the MTFa, R_{84}, SQF, Strehl ratio and RMS of wavefront aberration for their through-focus behavior in this chapter. (Note that different focal plane corresponds to different amount of W_{20}.) This (we hope) will help to identify a metric which well defines the best focal plane the eye will choose through accommodation. We used the commercial software package WISP of Wyko Corporation (using 128x128 array to represent pupil function, PSF and MTF) to calculate values of MTFa, R_{84}, Strehl ratio and RMS of wavefront aberration for different defocus value W_{20}. We restrict our attention to primary aberrations here, since practically all the psychophysical investigations have also been thus restricted. Furthermore, only even aberrations such as spherical (W_{40}) and astigmatism (W_{22}) are considered, since the odd aberrations such as coma (W_{31}) do not change the plane of best focus.

There are two questions to answer: (1) the location of the best focal plane for a specific metric (i.e. what value of W_{20} maximizes or minimizes the value of a metric); (2) how well it is defined (i.e. how sharp the maximum or minimum is).

Calculations performed with WISP show that:

(1) In the case of astigmatism, if we restrict the MTF to be at 45^0 only, then all metrics agree that the best focus is at the "circle of least confusion" (i.e. $W_{20} = -W_{22}/2$), for values of $W_{22} \leq 0.7\lambda$. As the astigmatism increases, the PSF at the S (sagittal) and T (tangential) foci shows a double hump, so the Strehl ratio is no longer well defined because the PSF does not have a single central maximum. For example, with one wavelength of astigmatism ($W_{22} = 1\lambda$), the PSF at the S or T focus has a central value of 0.18, but a maximum value of 0.24 on either side of the center. This is the reason for the

Fig 3.1 Normalized values of image quality metrics as a function of defocus for $W_{22} = 1\lambda$



(Note: RMS refers to the RMS wavefront aberration. All metrics, except R84 and RMS, are normalized to unity at their maximum. R84 and RMS are minimum at the optimum focus, so the inverse normalized values are plotted.)

strange behavior of the Strehl ratio shown in Fig 3.1. In that figure, the normalized value of each of the metrics was plotted as a function of W_{20}. Except for the Strehl ratio, the various metrics exhibit the same behavior even for larger values of W_{22} and they predict the best focus is still at the "circle of least confusion" (i.e. $W_{20} = -W_{22}/2$). The conclusion is that, in the presence of astigmatism, any metric except the Strehl ratio can be used to determine the best focus. Of the remaining metrics, the SQF is the least suitable, because it is quite insensitive to defocus.

(2) Spherical aberration behaves differently from astigmatism, in that the various metrics predict different optimum foci. Figures 3.2, 3.3, and 3.4 show this effect for 1, 2, and 3 λ of W_{40} In all cases, the defocus value that minimizes the wavefront variance (the so-called "best diffraction focus") is for $W_{20} = -W_{40}$. For 1λ and 2λ of W_{40} , almost all metrics predict approximately the same best focus at the best diffraction focus except for R_{84}. As might be expected, the differences between the different metrics are large only for larger amounts of aberration (greater than about 2λ), as indicated in figure 3.4. When spherical aberration increases to around 3λ, the Strehl ratio is no longer a good metric to predict the optimum focus because it shows double peaks through focus. Of the remaining metrics, the SQF is still the least suitable, because it is insensitive to defocus.

Fig 3.2 Values of image quality metrics as a function of defocus for $W_{40} = 1\lambda$ (normalization as in Fig 3.1)

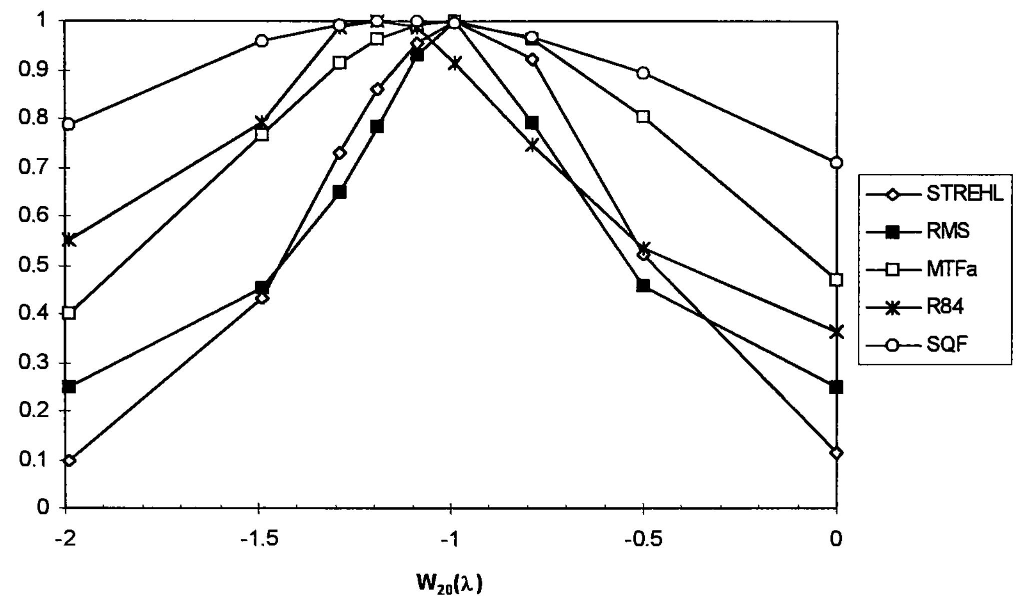


Fig 3.3 Values of image quality metrics as a function of defocus for $W_{40} = 2\lambda$ (normalization as in Fig 3.1)

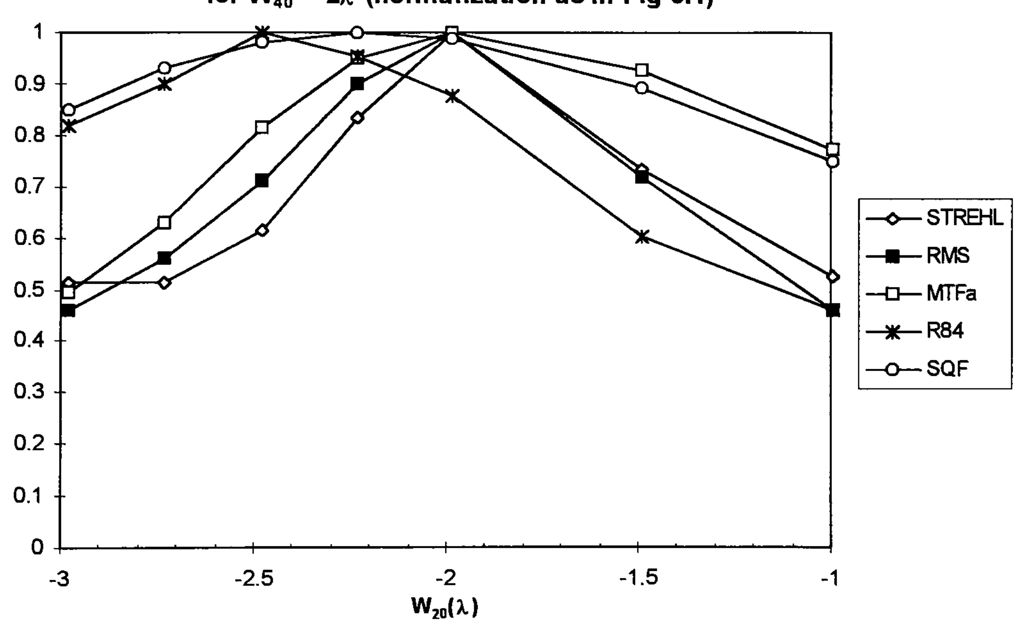
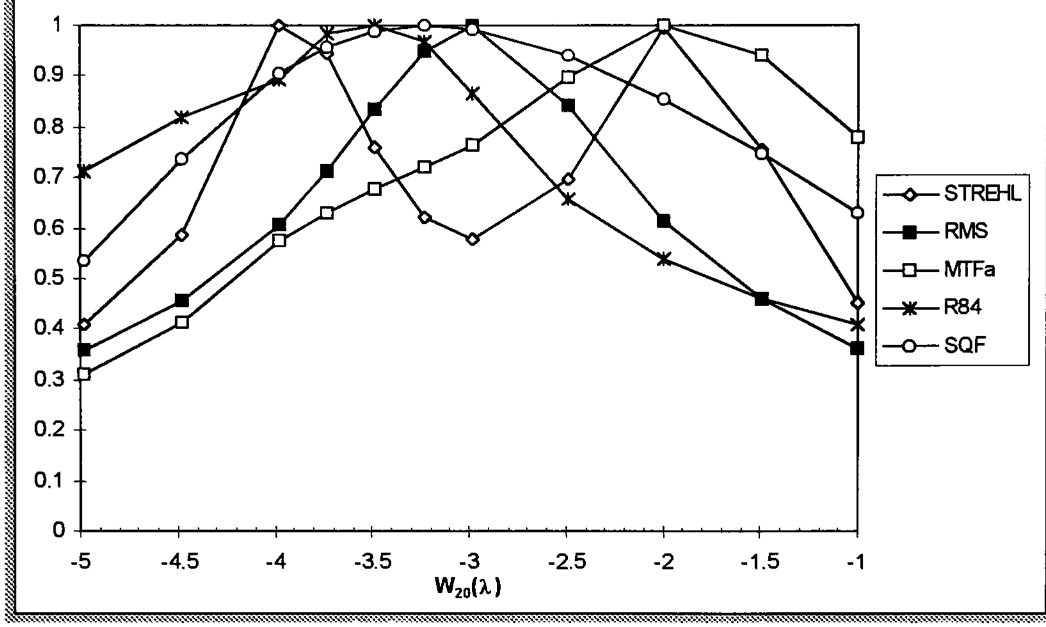


Fig 3.4 Values of image quality metrics as a function of defocus for $W_{40} = 3\lambda$ (normalization as in Fig 3.1)



From the above discussion, we can conclude that spherical aberration is the one which can separate different metrics in determining the optimum focus. Specifically, we see from Fig 3.4 that 3λ of spherical aberration will produce a substantial, easily measurable difference in the optimum focus location. Therefore, a viewing apparatus can be constructed which presents this amount of spherical aberration with negligible amounts of other aberration. For the right pupil size, the difference in focus between the different criteria (specifically R_{84} and MTFa) can be made to exceed 1 diopter. Therefore, it should be possible to assess the subjective performance by having the optimum focus for one or more of the objective metrics be outside the accommodative range of the observers. Then upon refocusing the eyepiece, this optimum focus can be brought within the observers' accommodative range, and any changes in subjective performance noted. This kind of experiment will help to identify a metric between R_{84} and MTFa which better describes the focal plane chosen through accommodation. In fact, such an experiment has been conducted by K. Lyons *et al*[22] They found that, of the three observers participated in the focusing experiment with paralyzed accommodation using telescopes with predominantly 1λ, 2λ and 3λ of spherical aberration, one consistently preferred the plane of optimum MTFa, another preferred a plane between the optimum MTFa and the paraxial focus, and the third was between the two.

Chapter 4

Correlation Results of Image Quality Metrics and Subjective Performance

As we discussed in chapter 1, there are some conflicting conclusions about the correlation of image quality metrics such as MTF, Strehl ratio, etc. and subjective performance in the past work. Here we examine some investigations by attempting to fit the experimental data with the various metrics. Our purpose is to determine if these conflicting conclusions can be reconciled. According to the targets used, we can separate the investigations into two categories as follows; those using one-dimensional targets such as gratings, edges and bar targets, and those using targets with two-dimensional details.

4.1 Correlation results of image quality metrics and subjective performance for two-dimensional targets

First we examine the results of Haig and Burton.[7] They used two types of targets with two-dimensional detail: a picture of a tank and a pseudorandom pattern. In order to determine the minimum amount of aberration that causes a just-noticeable difference

(JND) in image quality, they used a video monitor to simulate the effect of the aberrations. This method is especially useful when the aberrated PSF is known for aberrations that are not expected to interact with accommodation, such as coma,[4, 13] or for which the accommodative response has been reasonably well established, such as astigmatism[6] or for very small aberrations just like in Haig and Burton's work[7] where the eye is known to accommodate very close to paraxial focus. They presented the aberrated and unaberrated images side by side with the aberrated image generated by convolving the target with an aberrated PSF, and the unaberrated image generated by convolving the target with the diffraction limited PSF. A 2 mm artificial pupil was used in front of the eye to view the screen. Two-alternative forced choice was adopted as the psychophysical technique, and the threshold aberration value was taken as the 75% discrimination level.

Table 4.1 lists the aberration values (including some higher-order aberrations) causing one JND in image quality found in Haig and Burton's study. We can see from Table 4.1 that the amounts of the various aberrations are different, but the subjective effect is similar in all cases (one JND). Thus a successful image quality metric should have approximately the same value in all cases. Haig and Burton concluded that the Strehl ratio is the desired metric, but did not appear to have examined other metrics. In order to explore other possibilities, we computed the corresponding values of the following image quality metrics for each aberration condition in Table 4.1 : (1) Strehl ratio, (2) R_{84}, (3) MTFa (5 to 20 cycle/deg), averaged over azimuth by taking one-quarter of the sum of the S, T, $+45^0$ and -45^0 MTFa values, by using a 512x512 FFT (fast Fourier transform) program.

The 512x512 FFT program was developed by us which can calculate the above three metrics as well as MTFv with relatively higher accuracy than many commercial software packages. The program is attached in Appendix A and its accuracy was tested in different ways as follows: (1) by the analytical expression for the Airy disk, (2) by the analytical expression for the diffraction-limited MTF, (3) by the analytical expression for the encircled energy of the Airy disk,[15] (4) by the analytical expression for the MTF in the presence of defocus,[16] and (5) by the encircled energy function for rotationally symmetric aberrations, which was computed independently in polar coordinates by integrating along the radius. In all cases, the agreement was better than 1%. The details are given in Appendix A.

The calculation results of Strehl ratio, R_{84} and MTFa by the 512x512 FFT program for all the aberration values causing one JND were normalized by dividing the corresponding metrics' values in the absence of aberrations and also shown in Table 4.1.

In Table 4.1 we added tilt $W_{11} = -2/3\ W_{31}$ and $W_{11} = -1/2\ W_{51}$ (conditions for minimum variance of wavefront aberration) to W_{31} and W_{51} aberration cases.

Table 4.1 also shows the maximum and standard deviation from the mean for each metric, both normalized through division by the mean. We can see that the MTFa is better than the other two metrics since it has the least maximum and standard deviation. All metrics, however, fit this set of data well. (A 12% error is not considered large in psychophysical experiments of this sort.)

Table 4.1 Values of metrics corresponding to one JND in image quality (R_{84} and MTFa are normalized by dividing their aberration-free value)

aberration (λ)	Strehl ratio	R_{84}	MTFa
$W_{20} = 0.200$	0.875	1.32	0.962
$W_{40} = 0.190$	0.880	1.35	0.957
$W_{60} = 0.210$	0.869	1.41	0.947
$W_{80} = 0.245$	0.846	1.52	0.933
$W_{22} = 0.268$	0.838	1.30	0.964
$W_{42} = 0.305$	0.845	1.36	0.954
$W_{62} = 0.333$	0.854	1.40	0.949
$W_{31} = 0.395$*	0.918	1.25	0.963
$W_{51} = 0.350$**	0.904	1.25	0.954
mean	0.870	1.35	0.954
max. deviation	5.5%	12.5%	2.2%
std. deviation	3.1%	6.4%	1.0%

*: with $W_{11} = -2/3\ W_{31}$. **: with $W_{11} = -1/2\ W_{51}$

Since the results of Haig and Burton[7] show significant variation among the two observers and the two targets quoted, we were concerned if the conclusions drawn above are affected by the averaging across observers and targets. To confirm this, we then computed all the image quality metrics for the various combinations of observer and target. The results show that the worst-case condition gave the following normalized standard deviations: (1) Strehl 5.1%, (2) R_{84} 8.9%, and (3) MTFa 1.6%. Comparing these values with the last row of Table 4.1, we see that the conclusions hold independent of target and observer.

4.2 Correlation results of image quality metrics and subjective performance for one-dimensional targets

The second experiment we examined used a real telescope with several astigmatism and defocus values.[6] Three bar resolution targets and sinusoidal gratings were used in the experiment. The eye pupil was the limiting pupil in this experiment. We assumed the 2.8 mm for the diameter of the eye pupil since it was expected from the field luminance and also gave a good fit to the data. The sinusoidal contrast sensitivity was determined through the random double staircase technique, and three-bar resolution measurements were also taken at various contrast levels.

Of all the different aberration combinations of Ref. 6, we examine here only the ones with small amounts of astigmatism which are 0.28λ, 0.59λ, and 0.8λ; this is because other conditions involved either positive (hyperopic) image vergences, or very large amounts of astigmatism (more than 2.5λ to 3λ). In the first case, the optimum focal plane is not accessible to the normal eye, which leads to unpredictable accommodative responses; there is no image quality metric applicable to this situation. In the second case, the amounts of astigmatism were so large that the MTF dropped to zero around 12 cycle/deg (refer to Fig 2.1) which makes MTFa no longer a valid image quality metric. The reason is, when the MTF drops to zero within or near the region of the MTFa, no correlation should be expected between the MTFa and contrast sensitivity or resolution experiments. A zero crossing of the MTF at one focal plane does not mean that a grating of the corresponding spatial frequency is invisible; the eye can reaccommodate at another plane, where the contrast is not zero (different MTF). Therefore the response of the eye, as examined with sinusoidal gratings, would be expected to be a composite of different focal planes for different spatial frequencies. In such a case, then, the MTF at any single focal plane cannot describe the subjective response adequately. This problem does not exist in the restricted aberration range we examine here (0.28λ, 0.59λ and 0.8λ astigmatism), because there the effect of defocus is similar for all spatial frequencies within or close to the MTFa limits. That is, there exists a best focal plane that maximizes simultaneously the MTFa and the contrast of the individual spatial frequencies used in collecting the subjective data (which ranges from 6 cycle/deg to 28 cycle/deg in Ref. 6).

For the three aberration conditions (0.28λ, 0.59λ, and 0.8λ of astigmatism) of Ref. 6, we calculated the subjective resolution degradation and the subjective contrast degradation (averaged over all frequencies from 6 to 28 cycle/deg); these are defined as the ratios of the resolution and threshold contrast, respectively, to the corresponding values in the aberration free case. The subjective degradation results are shown in Table 4.2.

We noted that in all cases of Ref.6, the line targets were presented at 45^0 azimuth; therefore, in computing the objective metrics, we assumed that the observers accommodated halfway between the two astigmatic lines (i.e. $W_{20}=-W_{22}/2$). Then we used the 512x512 FFT program to calculate the Strehl ratio, R_{84} and MTFa (at 45^0 azimuth, from 5 to 20 cycle/deg frequency range) for the three aberration conditions (0.28λ, 0.59λ, and 0.8λ astigmatisms plus defocus $W_{20}=-W_{22}/2$ for each case). The results of these objective metrics were normalized by dividing the corresponding values in the absence of aberrations and shown in Table 4.2. We performed linear regression between the subjective contrast and resolution degradation and the corresponding values of each of the metrics. Table 4.3 shows the correlation coefficients (R^2) obtained.

Table 4.2 Subjective contrast degradation (CD), resolution degradation (RD) and values of objective metrics for the experiment of Ref.6

W_{22} (λ)	Contrast Degradation (CD)	Resolution Degradation (RD)	Strehl	R_{84}	MTFa @45^0
0.28	1.00	0.92	0.879	1.232	0.976
0.59	0.80	0.78	0.560	1.434	0.899
0.80	0.60	0.72	0.341	1.712	0.822

Table 4.3 Correlation coefficients (R^2) between CD and RD of Table 4.2 and the image quality metrics

	MTFa @45^0	R_{84}	Strehl
CD	1.00	0.99	0.99
RD	0.95	0.90	0.98

From Table 4.3, we can see that all the metrics fit this set of experimental data very well. It must be noted that since there are only three points in the regression, the statistical significance of the results is not great. However, this experiment serves to illustrate the point that when only one aberration type is involved, and over a restricted range of values, almost any image quality metric may give good correlation results.

The third experiment we examined here is the experiment of Mouroulis and Zhang[8] They simulated aberration effects on a video monitor in a way similar to that of Haig and Burton[7] The aberrations they examined were coma, astigmatism and their combinations and the magnitude of the aberrations was increased to include less-well corrected systems (see Table 4.4). The targets presented were sinusoidal gratings, a narrow bar and an edge. A 3-mm artificial pupil was used. Five observers participated and their results were sufficiently similar to permit meaningful averaging. In the experiment, two targets were presented side by side, each degraded at random by convolving with one of the aberrated or unaberrated PSFs. The observers' task was to state which target was more degraded (e.g., less contrast for the sinusoids, more blur for the bar or edges). This paired comparison technique resulted in a relative scale that characterized the perceived degradation of the targets.

Table 4.4 Aberration conditions used in the experiment of Ref. 8.

W31 (λ)	W22 (λ)
0.00	0.00
0.62	0.00
0.00	0.36
0.62	0.36
0.36	0.62
1.00	0.00
0.00	0.50
1.00	0.50
0.50	1.00

In all cases, $W_{11} = -2W_{31}/3$ and $W_{20} = -W_{22}/2$.

The aberration conditions used in the experiment of Mouroulis and Zhang[8] are listed in Table 4.4. Mouroulis and Zhang[8] calculated the values of Strehl ratio, R_{84} and MTFa (for 6 to 24 cycle/deg spatial frequency range) and gave the correlation results with subjective scale in Ref.8. We quote their results in Table 4.5 for the correlation coefficients (R^2) between the subjective scale and the different metrics, for the tangential target orientation that caused maximum image degradation. Other orientations were also briefly examined by Mouroulis and Zhang[8]. The correlation coefficients between the subjective scale and the MTFa at the 45^0 and sagittal (*S*) target orientations are given in Table 4.6. The results of this experiment were that the MTFa gives best correlation for all targets and all orientations. The other metrics do not depend on orientation and cannot be expected to correlate in all cases. However, for tangential orientation, R_{84} gave very good correlation.

Table 4.5 Correlation coefficients (R^2) for the various targets and objective metrics for tangential target orientation (quoted from Ref.8)

	MTFa	Strehl	R_{84}
sinusoidal gratings	0.99	0.67	0.97
edge	0.96	0.57	0.96
narrow bar	0.99	0.53	0.98

Table 4.6 Correlation coefficients (R^2) for the MTFa at the 45^0 and sagittal (*S*) target orientations (quoted from Ref.8)

	MTFa @ 45^0	MTFa @ sagittal
sinusoidal gratings	0.99	0.99
edge	0.86	0.96

Finally, we review the results of Giles,[4] who used real telescopes and a variety of aberrations, including spherical, coma, and astigmatism. Seven observers performed a contrast sensitivity experiment using square-wave gratings and "noise" (empty field) presentations using certainty-ratings technique. In addition, a standard three-bar resolution target was used at different contrasts. Giles calculated a contrast and a resolution degradation ratio, measured in the presence and absence of aberrations, and then multiplied the unaberrated MTF by the resulting degradation ratios for each spatial frequency. Reasonably good agreement was obtained between the curves thus obtained and the aberrated MTF curves computed at the plane of minimum wavefront aberration variance (i.e. the diffraction focus). However, Giles does not appear to have considered the correlation with any MTF integral or other metrics. To do that, we need to average the degradation ratios of Ref.4 across all spatial frequencies, so as to obtain a summary subjective degradation measure. This was done for both the grating and the three-bar targets used in Ref.4; the results are shown in Table 4.7, together with the corresponding aberration values. The symbols T for the three-bar degradation and D for the contrast degradation are the same as in Ref. 4.

Table 4.7 Aberration conditions and averaged subjective degradation ratios for the experiment of Ref. 4.

condition	W_{40} (λ)	W_{31} (λ)$^{\alpha}$	W_{22} (λ)	T (three-bar resolution) 5-18c/deg	D (grating detection) 5-14c/deg
1	1.3	1.0	0.15	0.78	0.81
2	0.18	1.07	0.21	0.61	0.70
3	0.1	0.64	0.93	0.59	0.74
4	2.6	1.35	0.11	0.47	0.52
5	0.06	0.69	1.84	0.30	0.33
6	-0.14	2.6	0.53	0.31	0.29

$^{\alpha}$ $W_{11} = -2W_{31}/3$ in all cases.

We then computed the MTFa, the Strehl ratio, and R_{84}, for the aberration values provided in Ref.4, using the 512x512 FFT program. The following conditions were observed during the computation of the objective metrics.

1. The MTFa was limited to the frequency range of the subjective measurements. This was 5 to 14 cycle/deg for grating detection, and 5 to 18 cycle/deg for three-bar resolution.
2. The astigmatism was oriented at 45^0 relative to the bars, and the coma was oriented across the bars. Therefore, in computing MTFa, the 45^0 MTF was taken for those cases where astigmatism predominates, and the worst MTF was chosen for all other cases.
3. We allow the focal plane (W_{20}) to vary in order to optimize the value of each metric, rather than restrict the choice to the planes of minimum wavefront aberration variance as was done by Giles.
4. We ignored the 0.25λ of undercorrected spherical aberration that Giles assumed to be due to the eye. The main reason for doing so was that its inclusion did not change any of the objective values appreciably. Also, it is now known that the monochromatic aberrations of the eye suffer from considerable interobserver variation, as well as variation with the state of accommodation, and that they are not adequately described as simply third-order spherical.[19] The state of accommodation was not controlled in Giles's experiment. The observers were allowed to focus the eyepiece at will.

The correlations between *T*, *D*, and the corresponding MTFa degradation values (normalized with respect to aberration free case) are shown in Fig 4.1 and 4.2. In producing these figures, we have omitted condition 1 from Table 4.7, because we believe that to be an anomalous measurement, within the set of results of Ref. 4. The reason for this is as follows: Notice that condition 1 has essentially the same amount of coma as condition 2 but considerably more spherical aberration, yet it shows much better subjective responses. This is impossible to explain, even through cancellation of aberrations by the eye, since the eye aberrations are much smaller in magnitude (Giles used the value 0.25λ). A possible explanation is that the amount of coma for condition 1 was misprinted, or otherwise stated incorrectly in Ref. 4. Since the real amount is not known, meaningful objective metrics cannot be determined in this case.

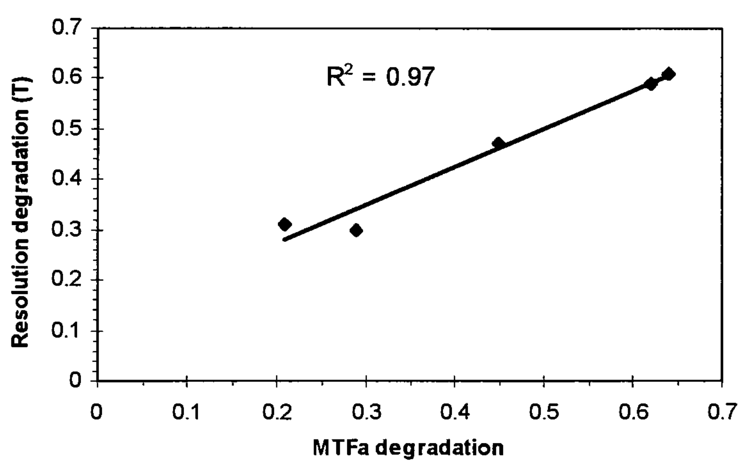
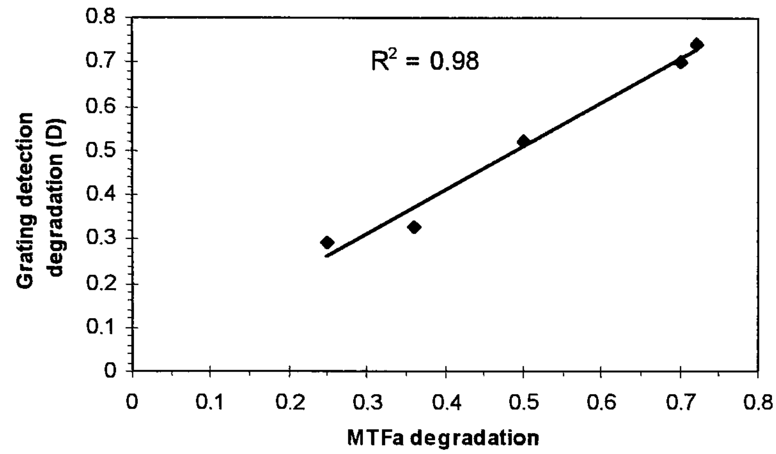


Fig 4.1 Correlation between the average subjective resolution degradation *T* of Ref. 4 and the MTFa degradation

Fig 4.2 Correlation between the average grating detection degradation *D* of Ref. 4 and the MTFa degradation



The other two metrics (Strehl and R_{84}) do not correlate very well with Giles's data. In Table 4.8 we summarize the correlation coefficients between *T* and *D* on one hand, and MTFa, Strehl, and R_{84} on the other.

Table 4.8 Correlation coefficients (R^2) between the subjective data of Ref. 4 and the objective metrics

	MTFa	Strehl	R_{84}
T (resolution degradation)	0.97	0.83	0.59
D (grating detection degradation)	0.98	0.71	0.69

Chapter 5

Focusing Experiment and Data Using Real Telescopes and DOG Target

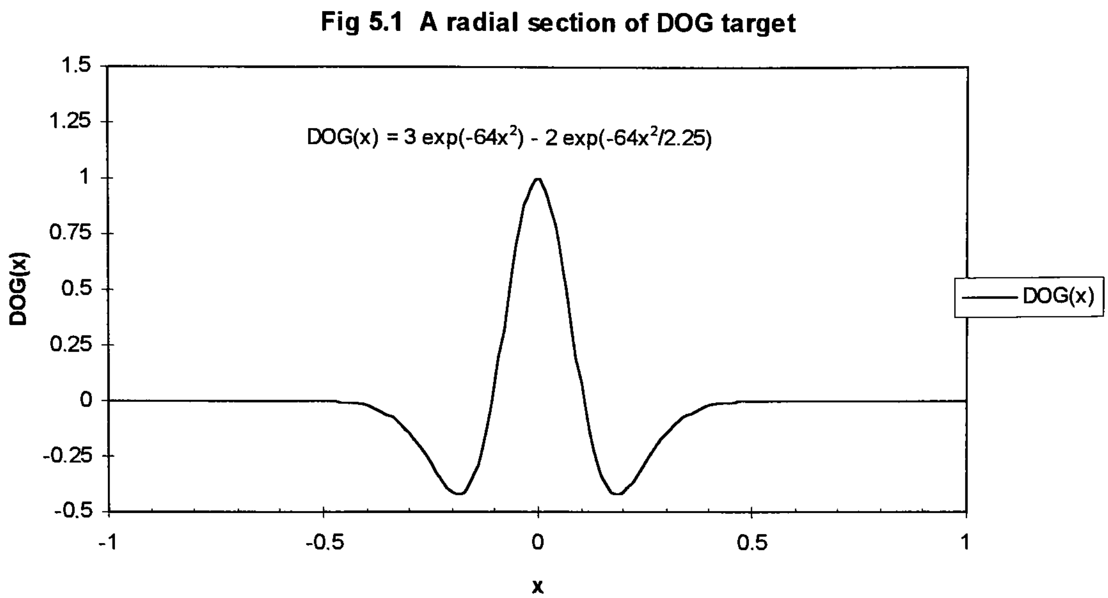
As we discussed in the Introduction, in order to test all metrics against two dimensional target as well as reduce the testing time and effort, we identified the DOG target. We performed the contrast sensitivity measurements using the DOG viewed through specially designed telescopes. In this chapter, we will describe: 1) the specific DOG target we used, 2) the aberration conditions of the real telescopes we tested, 3) the observers and the visual task, 4) objective metrics calculated through focus, 5) correlation results between subjective measures and the objective metrics.

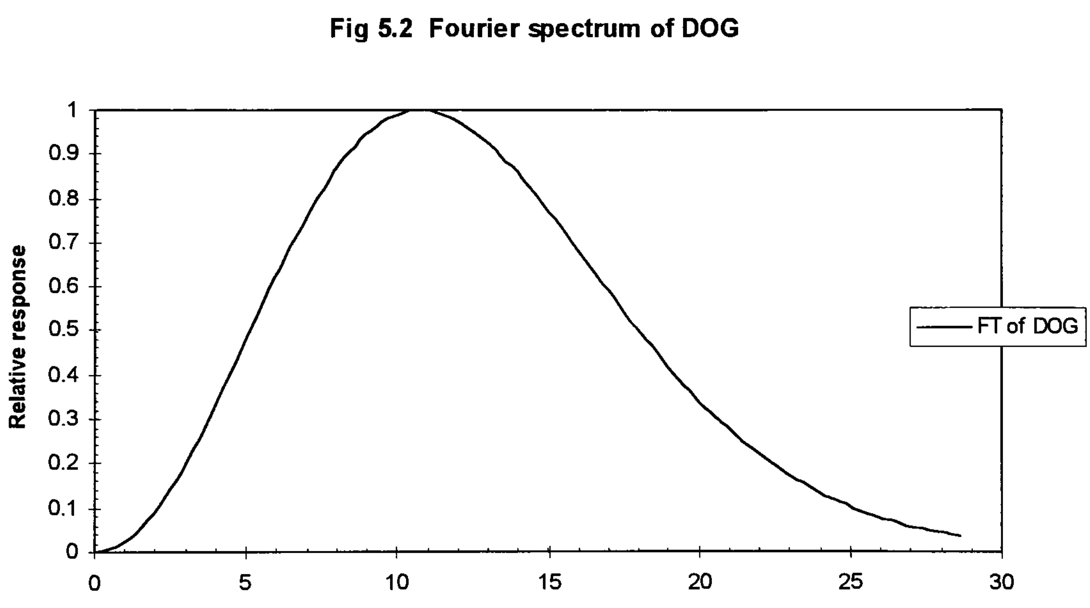
5.1 the DOG target

The mathematical expression for the DOG used in our experiment is

$$DOG(r) = 3\exp(-r^2/\sigma^2) - 2\exp(-r^2/2.25\,\sigma^2) \qquad \text{Eq.(5.1)}$$

where r is the radius from the center and the parameter σ is 1/8. A radial section of the DOG and its Fourier transform (calculated by Mathcad) are shown in Fig 5.1 and 5.2.





Note: when DOG's size (defined as the extent from x=-1 to x=1 for DOG(x) in Fig 5.1) subtends a viewing angle of 0.38^0, our calculation showed that the peak spatial frequency of DOG is 10.7 cycle/deg and the frequency band within 50% peak spectrum points is 5 - 18 cycle/deg.

Our calculation showed that, when the DOG's size (defined as the extent from x=-1 to x=1 for DOG(x) in Fig 5.1) subtends a viewing angle of 0.38^0, the peak spatial frequency of the DOG is 10.7 cycle/deg and the frequency band within 50% peak spectrum points is 5cycle/deg to 18 cycle/deg. This range of frequencies contains the most critical information about the state of correction of the instrument according to the discussion in section 2.4.

We then generated the DOG target on a high resolution (1280x1024) Mitsubishi HJ6905 color monitor which is a 19" nominally white video monitor, driven by a Pepper Pro-1280 display board from Number Nine Computer Systems at a 60 Hz noninterlaced vertical refresh rate and 64 kHz horizontal scan rate[21] The screen chromaticity coordinates were x=0.28, y=0.34. The pixel size was 0.296875 mm horizontally and 0.2734375 mm vertically. The computer program to generate the DOG target is attached in Appendix B. The monitor is able to present 256 discrete levels of luminance, the brightest being defined as 255 grey level and the darkest as 0 grey level. We defined the contrast of the DOG target as $(L_{max} - L_{min}) / (L_{max} + L_{min})$, where L_{max} and L_{min} were the luminances of the monitor corresponding to the maximum and minimum values of the DOG(r) function in Eq.(5.1). A contrast reduction was obtained by multiplying Eq.(5.1) with a constant less than one. For different contrast settings in our experiment, the average luminance of the DOG target was maintained to equal to that of the DOG boundary and the surround which was 128 grey level of the monitor. The actual luminance for 128 grey level of the monitor is 25 candela/m^2 which results in the corresponding eye pupil size of ~3.5 mm[24] The luminance profile on the monitor for the DOG with contrast of 0.1 is shown in Fig 5.3.

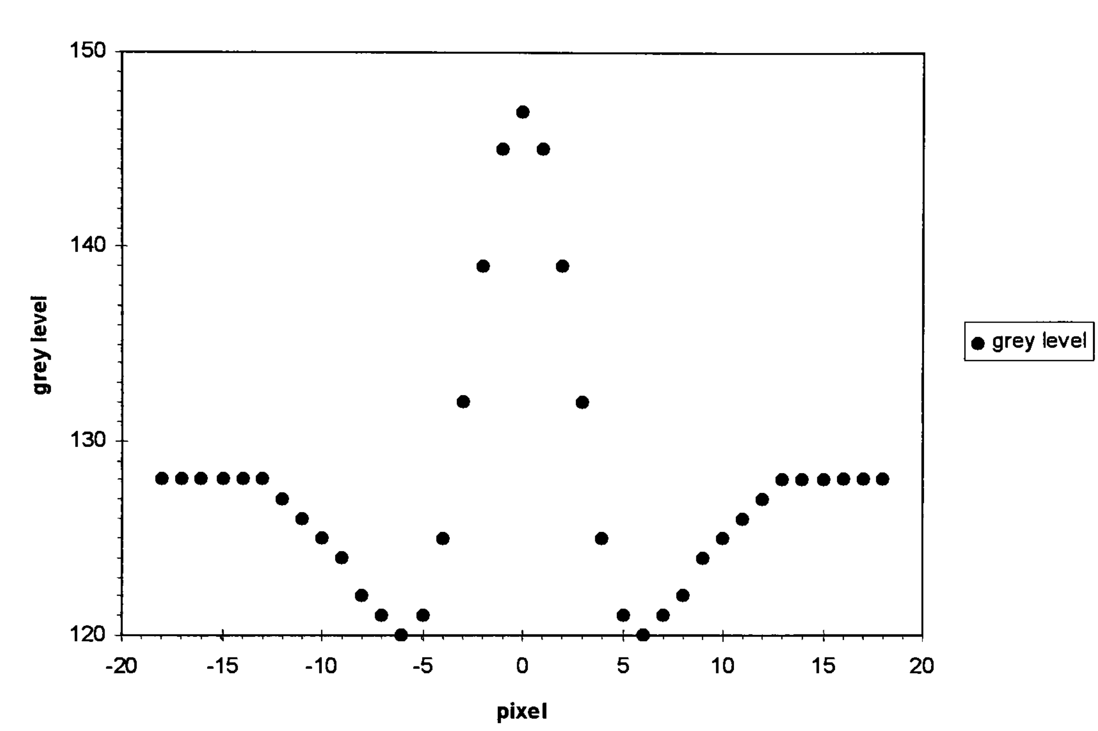


Fig 5.3 Luminance profile of the DOG target with contrast of 0.1

5.2 The aberration conditions of the telescopes

The observers viewed the targets through three specially designed Keplerian telescopes[22]. The telescopes were assembled from six off-the-shelf achromatic doublets (of known design prescription) in precision-machined barrels that provided for good alignment and replacement. An aperture stop (as the entrance pupil) can be attached to the telescope barrel with or without a 5cm extension tube. The aberration values of the telescopes were measured on a Twyman-Green interferometer illuminated with a He-Ne laser at 632.8 nm and the resultant fringe patterns were analyzed with commercial video fringe interpretation software. The measured aberration results showed good agreement with the design prescriptions. The telescopes had an exit pupil of approximately 3mm diameter, which provided the limiting pupil for the telescope/eye system.

Being constructed out of three pairs of doublets, the three telescopes had three slightly different angular magnification values. While viewing the target from around 7.1 meters distance, we adjusted the size of the DOG on the monitor to make it subtend a viewing angle of 0.38^0 in the eye space for telescopes with different magnification, thus making the peak spatial frequency of the DOG at 10.74 cycle/deg and the frequency band within 50% peak spectrum points from 5 cycle/deg to 18 cycle/deg throughout the entire experiment.

Different aberration conditions can be obtained from these three telescopes by controlling the orientation of the lenses, the entrance pupil location and the viewing condition. The entrance pupil was either at the objective lens or 5cm in front of the objective; in both cases, more than adequate eye relief is obtained. Predominantly spherical aberration (W_{40}) can be generated on-axis, but in order to test coma (W_{31}) and astigmatism (W_{22}), off-axis viewing was also included in the experiment. Table 5.1 shows the geometrical parameters for the various viewing conditions and the corresponding resulting aberration conditions. These aberrations were all third-order; higher orders were negligible. Telescopes no. 0, 1, 4, 5, 6 were all constructed from the same two lenses, the differences in aberration arising from the orientation of the lenses, the entrance pupil location as well as the viewing condition. For example, telescope no. 1 is obtained by turning around the objective lens of telescope no. 0. As can be seen from Table 5.1, the first condition (telescope no. 0) is essentially unaberrated.

Table 5.1 Geometrical parameters and aberration conditions for the telescopes (aberration values in wavelengths @550nm)

number	magnification	field angle	entrance pupil location	W_{40} (λ)	W_{31} (λ)	W_{22} (λ)
0	3.1	0^0	5cm in front	0.0	0.1	0.0
1	3.1	0^0	5cm in front	1.2	0.0	0.0
2	3.0	0^0	5cm in front	2.2	0.05	-0.05
3	2.5	0^0	5cm in front	3.2	0.17	0.0
4	3.1	3.5^0	5cm in front	0.0	-0.27	0.7
5	3.1	6.0^0	5cm in front	0.0	-0.4	2.0
6	3.1	6.0^0	at the objective	1.2	-1.5	2.0

Note: aberration values less than 0.05λ are shown as zero for clarity.

5.3 The observers and the visual task

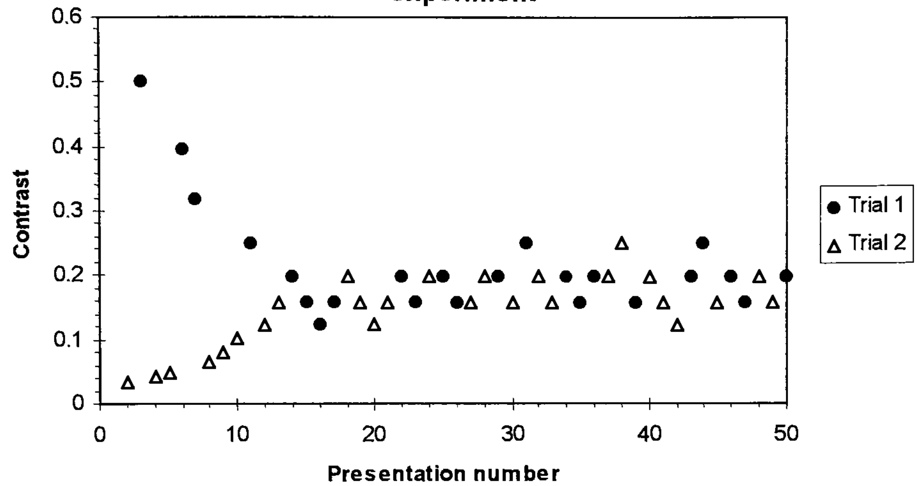
Three pre-presbyopic observers participated in the experiment. They all have normal color vision and corrected acuity better than 6/6. Their ages were 41 (male), 36 (male) and 30 (female).

There were two bars placed several degrees of angle away from the DOG target on the monitor. They were used to point toward the target and guide the observers to look at the right place within the otherwise unstructured field when the contrast of the target was near the threshold of the eye.

The random double staircase psychophysical technique was used to determine the contrast threshold of the DOG. In such an experiment, the observer responds "yes" if a target is perceived and "no" if it is not; the contrast for the next presentation is then lowered or raised by a fixed step. Observer bias is lessened by the random superposition of two series of presentations (one starting above threshold and the other below threshold, see Fig 5.4), so that the observer does not know which one (s)he is responding to. These series of presentations, or staircases, end up oscillating around the threshold value.

Fig 5.4 shows the typical session of a random double staircase run in our experiment. The starting contrast above threshold was around 0.5 for one series (Trial 1), and the starting

Fig 5.4 A typical session of random double staircase experiment



contrast below threshold was around 0.03 for the other (Trial 2). The contrast step was set so that

$$\Delta(\log(1/C)) = \log(1/C_2) - \log(1/C_1) = 0.1$$

where C is contrast, C_2 and C_1 are contrasts between two adjacent presentations in one staircase series. This resulted in larger steps for high contrast and smaller steps for low contrast so as to approach the threshold contrast more quickly. Each double staircase run consisted of at least 50 observations, of which the last 25 contrasts recorded were averaged to obtain the threshold. The observers controlled the experiment by inputting their own responses in the computer; the time for a single observation was 8 seconds, controlled by an auditory signal. A typical staircase run lasted around 10 to 15 minutes; no more than 3 runs were taken in a single day to avoid fatigue. All observers performed at least five practice runs; they all reported some difficulty in maintaining a constant threshold criteria. Two of the three who had participated previously in several sinusoidal contrast sensitivity experiments commented that they found the DOG experiment more difficult, which was reflected in the number of practice runs needed.

The observers were allowed to focus the telescopes at will, through the following process. Each observer determined the best focus three times for every aberration condition. The focus was then set at the average of the three positions, and remained fixed through a staircase run. In all cases, the three settings were close to each other, indicating a preference for a narrow range of image planes. The experimentally determined focus position was also checked against the calibration data of the telescopes, to ensure that it did not require extreme accommodation.

The raw results from the experiment comprise the threshold contrast for the various aberration conditions for each observer. Table 5.2 shows the raw threshold contrast (TC) data for observer 1. The standard deviation for these results (and for all observers) was approximately 15% of the threshold contrast. Table 5.2 also shows the raw contrast sensitivity (CS) data for observer 1, which is defined as the inverse of threshold contrast. A contrast sensitivity degradation is then obtained for each aberration condition by dividing all aberrated CS values by the unaberrated one (telescope no. 0). The data for the other two observers are similarly processed, but they are also multiplied by an additional normalization factor, given in the last row of Table 5.2. The reason to do this is as follows. Since our final aim is to produce correlation graphs, a better estimate of the subjective degradation may be obtained if the results of all three observers are averaged. However, in order to do that, we must first ensure that the degradation scale for the three observers is similar; thus we compute the average threshold over 6 aberration conditions for all observers, and then normalize the results with respect to a single observer (this does not affect the correlation). The two columns of Table 5.2 labeled "observer 2" and "observer 3" represent the contrast sensitivity degradation for observer 2 and 3 normalized relative to the average contrast sensitivity degradation of observer 1. The last column gives the average of the three observers on the same degradation scale (previous three columns). It can be seen from the data that all three observers showed the same trends and hence the averaging process is meaningful.

Table 5.2 Subjective data for the aberration conditions of Table 5.1

number	observer 1 TC	observer 1 CS	observer 1 CS degradation	observer 2 CS degradation (normalized)	observer 3 CS degradation (normalized)	average
0	0.08	12.50	1			
1	0.093	10.75	0.86	0.75	0.77	0.793
2	0.12	8.33	0.67	0.65	0.64	0.653
3	0.14	7.14	0.57	0.59	0.58	0.580
4	0.105	9.52	0.76	0.81	0.82	0.797
5	0.132	7.58	0.61	0.67	0.62	0.633
6	0.146	6.85	0.55	0.56	0.59	0.567
			(1)	(0.81)	(0.85)	

5.4 Objective metrics calculated through focus

We then computed the values of the following image quality metrics: MTFa, MTFv, R_{84} and Strehl ratio using the 512x512 FFT program and normalized them by dividing the nominally unaberrated values of telescope no. 0. In this case, we define the MTFa as the integral of the MTF between 5 and 20 cycle/deg, averaged over the sagittal, tangential, $+45^0$, and -45^0 azimuths. The MTFv is defined as the two dimensional integral of the MTF over the same frequency range. For a two dimensional target, the MTFv is more appropriate than MTFa, but it is not commonly computed by commercial software. With respect to the R_{84} normalization, we note that R_{84} increases with aberration, so the normalized values are greater than one.

In order to find the optimum value of individual image quality metric through focus, we computed each of them for all aberration conditions through focus (in step of 0.2λ or 0.1λ of W_{20}) and plotted the results in Fig 5.5, 5.6, 5.7, 5.8, 5.9 and 5.10. The optimum values of these metrics are given in Table 5.3.

Fig 5.5 Values of image quality metrics as a function of defocus for telescope no. 1

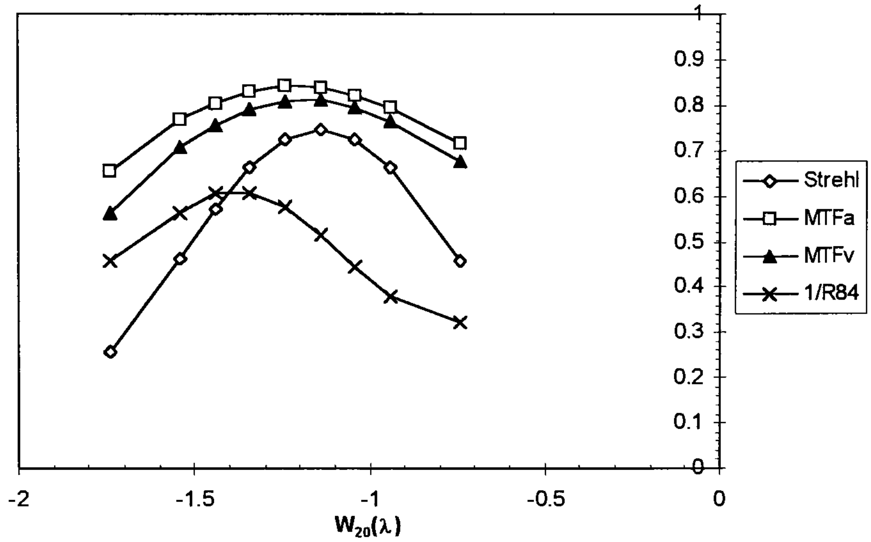


Fig 5.6 Values of image quality metrics as a function of defocus for telescope no. 2

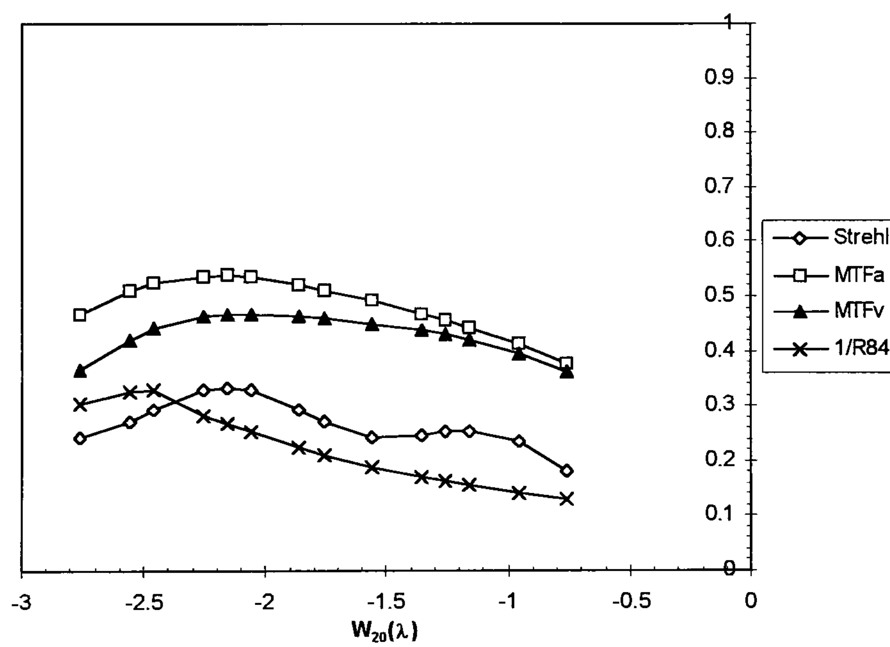


Fig 5.7 Valuesof image quality metrics as a function of defocus for telescope no. 3

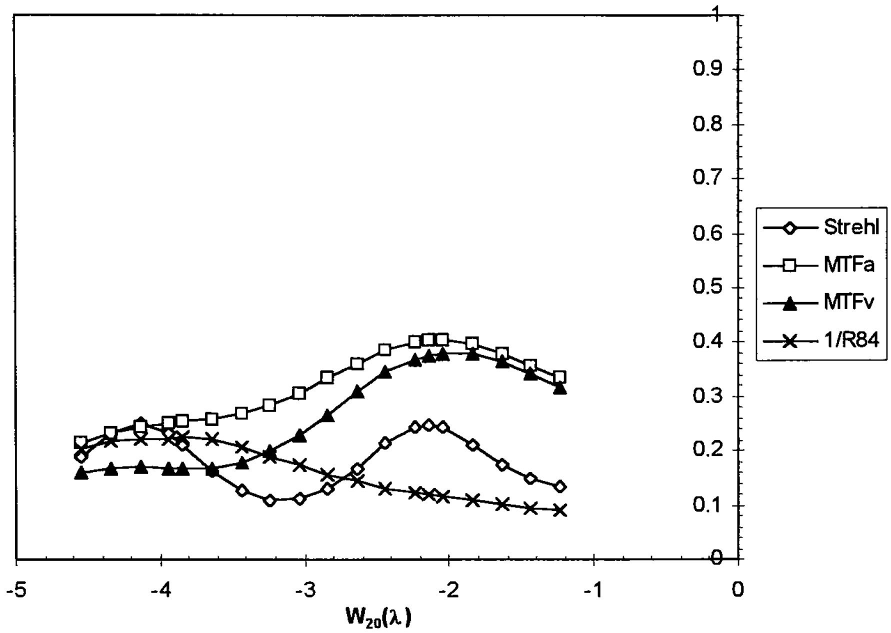


Fig 5.8 Values of image quality metrics as a function of defocus for telescope no. 4

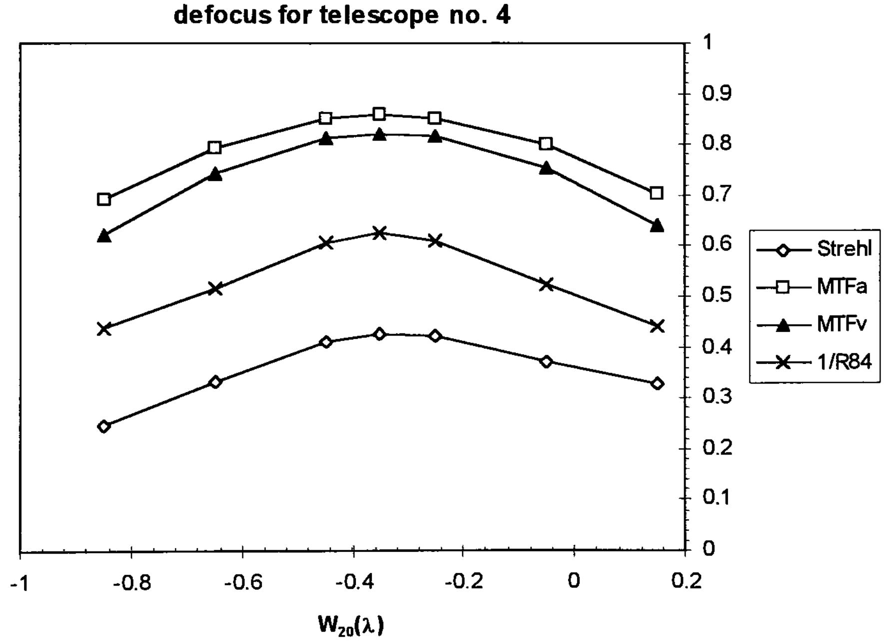


Fig 5.9 Values of image quality metrics as a function of defocus for telescope no. 5

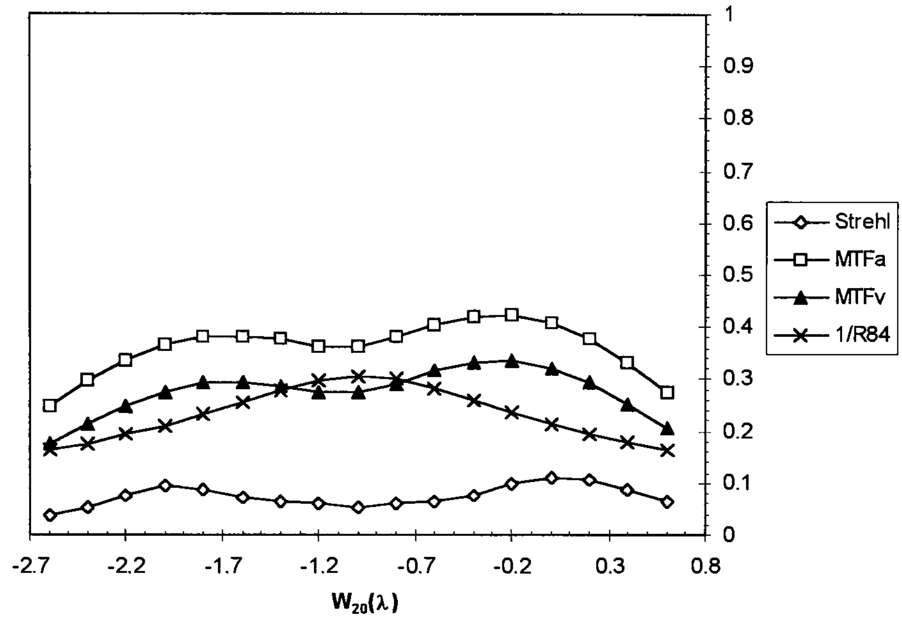


Fig 5.10 Values of image quality metrics as a function of defocus for telescope no. 6

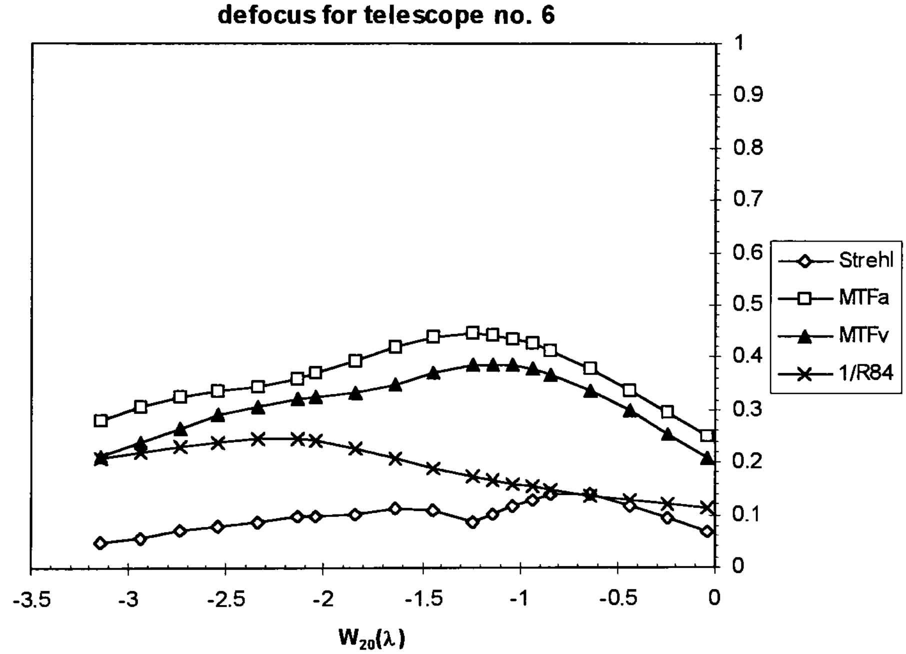


Table 5.3 Optimum values of MTFv, R_{84}, and Strehl ratio for aberration conditions of Table 5.1

number	Strehl ratio	MTFv (5-20cyc/deg)	R_{84}
0	0.995	0.997	1.1
1	0.747	0.81	1.647
2	0.334	0.47	3.038
3	0.252	0.381	4.454
4	0.426	0.822	1.603
5	0.112	0.334	3.259
6	0.14	0.388	4.072

5.5 Correlation results between CS degradation and the objective metrics

We then performed linear regression (least squares fit) for the averaged subjective data in Table 5.2 versus the normalized values of the four metrics (R_{84}, MTFv, MTFa, and Strehl ratio) which were obtained by dividing the metrics data for telescopes no. 1 to no. 6 with that of telescope no. 0 in Table 5.3. The correlation results are shown in Fig 5.11, 5.12 and 5.13. The correlation coefficient R^2 for the MTFa plot was 0.94. However, this plot is not shown here in favor of the MTFv, since MTFv is a more appropriate metric for this experiment. The MTFa, when defined as the average area under the MTF over several azimuths, is merely a way of approximating the MTFv without having to perform the two-dimensional integration. The MTFa over a single azimuth is the appropriate metric to use for targets with one-dimensional detail or rotationally symmetric aberrations, as in Ref. 8 and 22.

It can be seen from Fig 5.11, 5.12 and 5.13 that R_{84} gives good correlation, and the Strehl ratio does not. The MTFv correlation may be considered as barely acceptable. In order to better appreciate the statistical significance of these results, we show in Table 5.4 the 95% confidence limits for the value of R in each case. The lower limit represents the worst case correlation. These limits are far apart as a consequence of the small number of points.

Fig 5.11 Correlation between the average subjective CS degradation and normalized R_{84} for the aberration conditions of Table 5.1

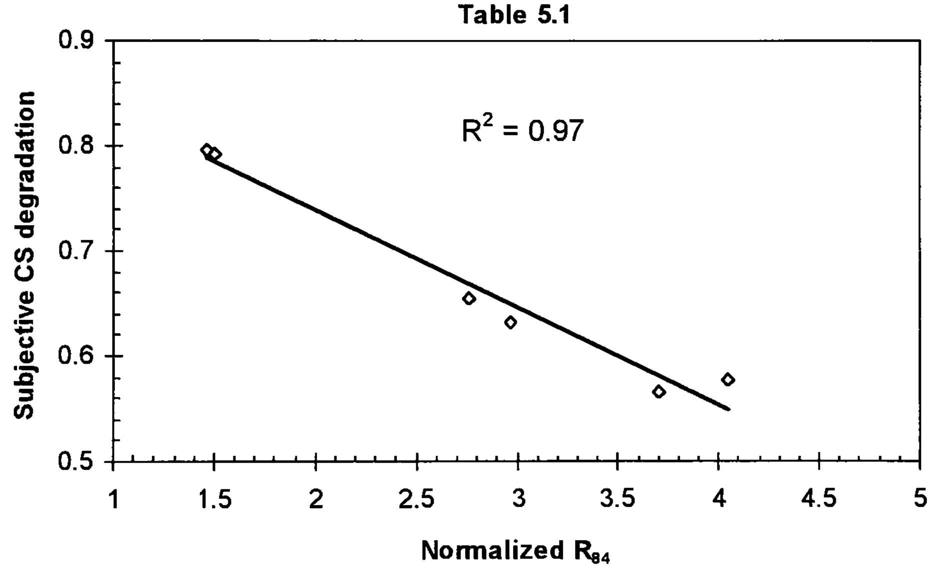


Fig 5.12 Correlation between the average subjective CS degradation and normalized MTFv for the aberration conditions of Table 5.1

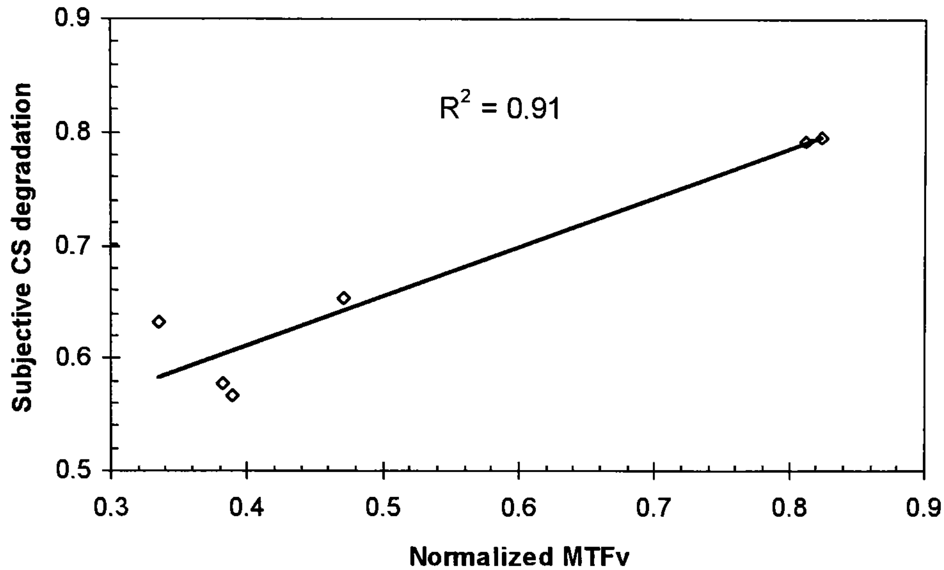


Fig 5.13 Correlation between the average subjective CS degradation and normalized Strehl ratio for the aberration conditions of Table 5.1

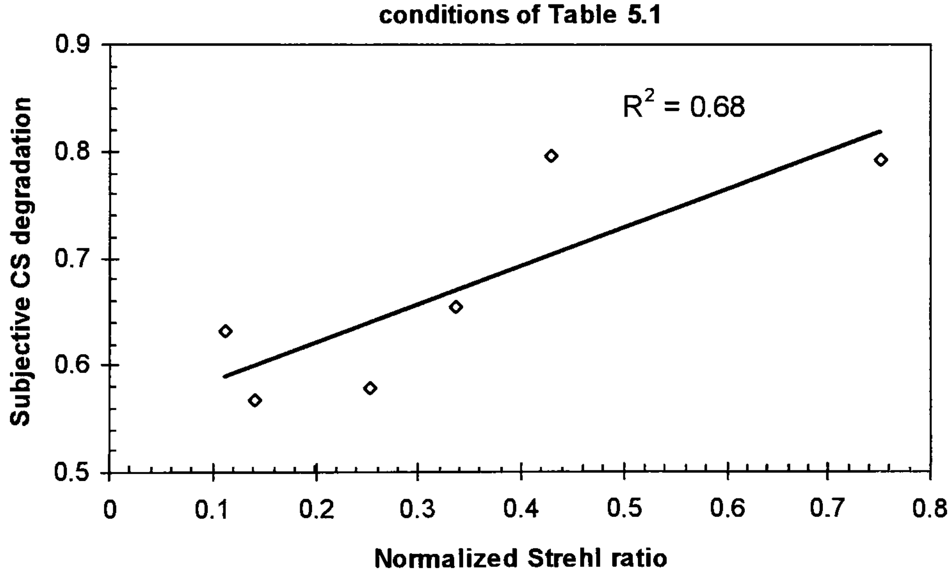


Table 5.4 95% confidence limits for the values of R

metric	R	upper limit	lower limit
R84	0.985	0.998	0.864
MTFv	0.954	0.995	0.630
Strehl ratio	0.825	0.980	0.040

Conclusions

(1) Regarding the optimum focus criteria, we have concluded from the examination of various objective metrics through-focus behavior that R_{84} and MTFa are the best candidates. We propose a focusing experiment using 3λ of spherical aberration that should further decide which of these two metrics better defines the focal plane the eye will choose through accommodation. The preliminary results from the focusing experiment by Lyons *et al*[22] indicate that the preferred focal plane in the presence of spherical aberration spans a range from the optimum MTFa focus to approximately halfway through the paraxial focus.

(2) Regarding the correlation between objective instrumental image quality metrics and subjective performance in resolution or contrast or related tasks, we have arrived at the following conclusions:

- for very small aberrations ($\approx 0.25\lambda$) of all types, including higher orders, for two-dimensional complex targets, using video simulation, all three metrics (MTFa, R_{84} and Strehl ratio) correlate well with subjective performance (derived from Haig and Burton's work[7]).

- for small amounts of astigmatism (0 to 0.8λ), for sinusoids and three-bar targets, using a real telescope, all three metrics correlate well with subjective performance (derived from Mouroulis's work[6]).
- for medium amounts (0 to 1λ) of astigmatism, coma and their combinations, for sinusoids, narrow bar and edge targets, using video simulation, MTFa and R84 correlate well with subjective performance (based on Mouroulis and Zhang's work[8]).
- for medium to large amounts of aberrations (0 to 2.5λ) of primarily coma and astigmatism and their combinations, for square-wave gratings, using real instruments, MTFa correlates well with subjective performance (derived from Giles's work[4]).
- for medium to large amounts of all primary aberrations (0 to 3λ) and their combinations, for two dimensional DOG target, using real telescopes, R_{84} correlates well with subjective performance. MTFv gives barely acceptable correlation (based on the current work).

Overall, we can see that MTFa (or MTFv) still correlates well in most cases. Although it cannot be considered to have passed clearly the test of the DOG's data, it also did not clearly fail. On the other hand, R_{84} showed good correlation in a limited number of cases, and may still be useful in characterizing aberration combinations and non-directional targets. Finally it seems safe to conclude that the usefulness of the Strehl ratio cannot be extended beyond the region of essentially diffraction-limited optical instruments.

Specifically, our current results on DOG targets show that the establishment of a successful and widely applicable image quality metric for visual instruments will require further experiments in which the influence of the target structure and the psychophysical task should be investigated in further detail. This first attempt (i.e. DOG) for alternative targets did not produce unequivocal results regarding the utility of the DOG target as a means of reducing the testing time and effort of contrast sensitivity tests without sacrificing too much information. Use of this target appears successful in handling both directional and non-directional aberrations irrespective of orientation. But the observers found the task to be relatively difficult. This is probably a consequence of the broad spatial frequency spectrum of the target which otherwise is a desirable attribute.

Lastly, it is important to remember that our conclusions drawn for small to medium amounts of aberrations should not be extended to larger amounts without experimental proof. This is because in the presence of very large aberrations the eye may not choose any appropriate focus. There is evidence[6] that when the target appears blurred everywhere, the accommodation can revert to its resting state, which can be far from any theoretical plane of best focus. In that case, drastically different methods of assessment will be required.

References

1. T. L. Williams, B. A. Leach, and B. J. Biddles, " A Workshop Instrument for Testing Binocular and other Sights Using the MTF Criterion", *Optics and Laser Technology* 4, 115-120 (1972).
2. N. P. Barton, "Application of the Optical Transfer Function to Visual instruments", *Opt. Acta* 19, 473 (1972).
3. B. A. Leach, "Ministry of Defence Experience of Introducing MTF into Production Testing of Optical Systems", *Opt. Acta* 22, 375 (1975).
4. M. K. Giles, "Aberration Tolerances for Visual Optical Systems", *J. Opt. Soc. Am.* 67, 634 (1977).
5. G. J. Burton and N. D. Haig, "Criteria for testing of afocal visual instruments", in *Assessment of Imaging Systems: Visible and Infrared*, T. L. Williams, Ed., *Proc. SPIE* 274, 191 (1981).
6. P Mouroulis, "On the correction of astigmatism and field curvature in telescopic systems", *Opt. Acta* 29, 1133 (1982).
7. N. D. Haig and G. J. Burton, "Effects of wavefront aberration on visual instrument performance, and a consequential test technique", *Appl. Opt.* 26, 492 (1987).
8. P. Mouroulis and H. Zhang, "Visual instrument image quality metrics and the effects of coma and astigmatism", *J. Opt. Soc. Am. A* 9, 34 (1992).

9. G. J. Burton and N. D. Haig, "Effects of the Seidel aberrations on visual target discrimination", *J. Opt. Soc. Am. A* 1, 373 (1984).

10. W. B. King, "Dependence of the Strehl ratio on the magnitude of the variance of the wave aberration", *J. Opt. Soc. Am.* 58, 655 (1968).

11. E. M. Granger and K. N. Cupery, "An optical merit function (SQF), which correlates with subjective image judgments", *Photogr. Sci. & Eng.* 16, 474 (1972).

12. W. N. Charman and J. Tucker, "Dependence of accommodation response on the spatial frequency spectrum of the observed object", *Vision Res.* 17, 129 (1977).

13. H. S. Coleman, D. G. Clark, and M. F. Coleman, "Interferometer studies of telescopic systems", *J. Opt. Soc. Am.* 37, 671 (1947).

14. W. N. Charman and A. Olin, *Photogr. Sci. & Eng.* 9, 385 (1965).

15. M. Born and E. Wolf, *Principles of Optics*, 5th ed., p.398, Pergamon. Oxford.

16. H. H. Hopkins, "The frequency response of a defocused optical system." *Proc. Roy. Soc. London Ser. A* 231, 91-103 (1955).

17. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, p.319, John Wiley & Sons. New York/Chichester/Brisbane/Toronto.

18. N. D. Haig and T. L. Williams, "Psychometrically appropriate assessment of afocal optics by measurement of the Strehl intensity ratio", *Appl. Opt.* 34, 1728 (1995).

19. W. N. Charman, "Optics of the human eye", in *Visual Optics and Instrumentation*, W. N. Charman, Ed., pp. 11-14, Macmillan (1991).

20. D. Regan, "A brief review of some of the stimuli and analysis methods used in spatiotemporal vision research", in *Spatial Vision*. D. Regan Ed., CRC Press (1991).

21. P. Mouroulis, T. Kim, and G. Zhao, 'Transverse color tolerances for visual optical systems", *Appl. Opt.* 32, 7089-7094 (1993).

22. K. Lyons, P. Mouroulis, and D. Cheng, 'Effect of instrumental spherical aberration on visual image quality", *J. Opt. Soc. Am.* A 13, 193-205 (1996).

23. A. Marechal, *Rev. Opt.* 26, 257 (1947).

24. J. W. Walsh, *Photometry*, (Constable, London, 1956).

25. F. W. Campbell, J. J. Kulikowski, and J. Levinson, 'The effect of orientation on the visual resolution of gratings", *J. Physiol.* 187, 427-436 (1966).

Appendix A.

512x512 FFT program and its accuracy tests.

The 512x512 FFT program contains three sub-programs named *wave.c*, *cheng.c* and *fourn.c* which need to be complied together using *makefile*. The final executable program *wave* was obtained after executing >*make wave* (>: Unix prompt sign). The program *wave* can calculate PSF and OTF (both MTF and PTF) of a circular pupil with wavefront Seidel aberrations of W_{11}, W_{22}, W_{20}, W_{31}, W_{42}, W_{40}, W_{51}, W_{62}, W_{60}, W_{80} and give the output in the format of 512x512 array. It also calculates Strehl ratio, MTFa, MTFv and R_{84}.

To run '*wave*', one should prepare *datafile* in the following format:

.1	2								
0	.1	0	0	0	0	0	0	0	0
.2	0	0	0	0	0	0	0	0	0
W11	W22	W20	W31	W42	W40	W51	W62	W60	W80

where the 1st number in the 1st line is the exit pupil radius (in ratio of 256); the 2nd number in the 1st line is the number of cases one wants to calculate; the 2nd and 3rd lines are two cases of aberrations in wavelength (λ), the corresponding terms of aberrations are listed in the last line (W11 W22 W20 W31 W42 W40 W51 W62 W60 W80) which is for reference only (not a necessity to run the program). Then one can run the program as the following:

prompt>*wave datafile*

The output (aberrations, strehl ratio, R_{84}, MTFa, MTFv) for each case will be automatically saved in a file named

datafile.out

Note that MTFa & MTFv is set from 0.05 to 0.21 normalized (to 1) frequency range which corresponds to 5 to 20cyc/deg for cutoff frequency of 95cyc/deg. One can change the frequency range for MTFa and MTFv as well as the ratio for encircled energy (e.g. to calculate R_{70} instead of R_{84} etc.) by modifying the corresponding parameters in the program.

wave.c

```
/* Module Title: psf&mtf stuff
 * programmer:  Xiaoxue Cheng\Kevin Lyons
 * intent: calculate strehl ratio, R84(or other e.g.R70),  mtfa & mtfv(over specified limits).
 * data input:  datafile [data] given on the command-line
 * data output:  strehl ratio, R838 mtfa and mtfv printed to [data].out.
 */

#include <math.h>
#include <stdio.h>
#include <string.h>

#define ABERR_NO 10  /* how many aberrations */
#define PI (double)4*atan(1)
#define N 512              /* array dimension */

/* global variables  */
double pupilr;         /* pupil radius as a ratio of N/2 */
double re[N][N];       /* real part for FFT      */
double im[N][N];       /* imaginary part for FFT */
char *aberra[]={ "W11=",  /*  tilt               */
           "W22=",  /* astigmatism        */
           "W20=",  /* defocus            */
           "W31=",  /* coma               */
           "W42=",  /* high order astig   */
           "W40=",  /* spherical          */
           "W51=",  /* high order coma    */
           "W62=",  /* high order         */
           "W60=",  /* 2nd order spherical */
           "W80="  };

/* function prototype */
void Instruct (void);
poly(double m[ABERR_NO], double *pupilvol);
extern void cheng( double pupilvol, double * strehl, double *r838, double *mtfa, double
*mtfv );
FILE *OpenFile(int argc, char *argv[]);
void ReadParameter(FILE *fp, int *no_of_case);
void ReadAbrr(FILE *fp, double w[]);
```

```
void InitArray(double w[]);
void PrintHeader(void);
void PrintResult( double w[],double strehl,double r838,double mtfa,double mtfv);

FILE *OpenFile(int argc, char *argv[])
{
   FILE *fp = NULL;   /* file pointer */

   if(argc == 1){
         printf("No file name specified on command-line, program terminates\n");
         exit(1);
   }
   else{
         fp = fopen(argv[1], "r");
         if( fp == NULL){
             printf("Data file does not exist, program terminates\n");
             exit(1);
         }
   }
   return(fp);     /* return file pointer */
}

void ReadParameter(FILE *fp, int *no_of_case)
{
    fscanf(fp, "%lf %d", &pupilr, no_of_case);
}

void ReadAbrr(FILE *fp, double w[])
{
    int i;

    for(i=0; i<ABERR_NO; i++)
        fscanf(fp, "%lf", &w[i]);

    return;
}

void InitArray(double w[])
{
    int i, t;

    for(i=0; i<ABERR_NO; i++)
```

```
        w[i] = 0.0;

    for(t = 0; t < N; t++){
        for(i = 0; i < N; i++){
            im[t][i] = 0.0;  re[t][i] = 0.0;
        }
    }
}

void PrintHeader(void)
{
    printf("\n\n\n radius = %lf\n", pupilr);
    printf("\n  aberration     strehl    R838    mtfa    mtfv  \n\n");
    printf(" -----------------------------------------------------------\n");
}

void PrintResult(double w[], double strehl,double r838,double mtfa, double mtfv)
{
    int i;

    for( i = 0; i < ABERR_NO; i++)
        if( w[i] != 0.0 )
            printf("   %s%.3f\n", aberra[i], w[i]);

    printf("%24.3f%9.3f%9.3f%9.3f\n", strehl, r838, mtfa, mtfv);
    printf(" -----------------------------------------------------------\n");
}

int main(int argc, char *argv[])
{
   FILE *fp;                          /* file pointer  */
   double w[ABERR_NO];     /* array to hold wave abrr. coeff. */
   double pupilvol, strehl, r838, mtfa, mtfv;
   double r838a=1., mtfaa=1., mtfva=1.; /* for aberration free case */
   int i, no_of_case;
   char s[40];

   Instruct();
   fp=OpenFile(argc, argv);

   strcpy( s, strcat(argv[1], ".out") );
   freopen(s, "w", stdout);
```

```
   ReadParameter(fp, &no_of_case);

   /* aberration free case */
   InitArray(w);
   poly(w, &pupilvol);
   cheng(pupilvol, &strehl, &r838a, &mtfaa, &mtfva );
   PrintHeader();
   PrintResult(w, strehl, r838a, mtfaa, mtfva);

   for( i = 0; i < no_of_case; i++){
      InitArray(w);
      ReadAbrr(fp, w);

      fprintf(stderr, "\nCase No %d\n", i+1);

      /* calculate pupil function */
      poly(w, &pupilvol);

      /* fft & psf & mtf calculation  */
      cheng(pupilvol, &strehl, &r838, &mtfa, &mtfv );

      /* print aberrations, strehl ratio & normalized  r838, mtfa, mtfv*/
      PrintResult(w, strehl, r838/r838a, mtfa/mtfaa, mtfv/mtfva);
   }
}

void Instruct (void)
{
   printf("   -----------------     Instruction    -------------------
   To run 'wave', you should prepare datafile in the following format:
``````````````````````````````````````````````````````````````````````````````

      .1   2
      0   .1   0   0   0   0   0   0   0   0
      .2   0   0   0   0   0   0   0   0   0
      W11   W22   W20   W31   W42   W40   W51   W62   W60   W80
``````````````````````````````````````````````````````````````````````````````

  where
      the 1st number in the 1st line is the pupil radius (in ratio of 256);
      the 2nd number in the 1st line is the number of cases you want to calculate;
      the 2nd and 3rd lines are two cases of aberrations in wavelength, the corresponding
terms of aberrations are listed in the last line (W11  W22  W20  W31  W42  W40  W51
W62  W60  W80) which is for your reference only (not a necessity to run the program).
```

```
   Then you can run the program as the following:
      prompt>wave datafile
   The output (aberrations, strehl ratio, R838, MTFa, MTFv)
   for each case will be automatically saved in a file named
      datafile.out
   Good luck!                    ......end
   Note: MTFa & MTFv is from 0.05  0.21 normalized (to 1) frequency range.
        (e.g. 5-20cyc/deg for cutoff=95cyc/deg)
   ---------------------------------------------------------------\n\n");
}

/* calculate pupil function */
poly(double m[ABERR_NO], double *pupilvol)
{
   int i, j;
   int sum=0.;
   double s, x = 2./N-1.;
   double t, y = 2./N-1.;
   double wabrr, temp;

   for(i = 0; i < N; i++){
        s = x/pupilr;
        for(j = 0; j < N; j++){
           if ((double)sqrt(x*x + y*y) <= pupilr){
              t = y/pupilr;
              temp = s*s + t*t;
              wabrr = m[0]*t + m[1]*t*t + m[2]*temp + m[3]*t*temp
                  + m[4]*t*t*temp + m[5]*temp*temp + m[6]*t*temp*temp
                  + m[7]*t*t*temp*temp + m[8]*pow(temp,3) + m[9]*pow(temp,4);
              re[i][j] = cos(2 * PI * wabrr);
              im[i][j] = sin(2 * PI * wabrr);
              sum++;
           }
           y = y + 2./N;
        }
        y = 2./N-1.;
        x = x + 2./N;
   }
   fprintf(stderr, "Energy before fft is %d\n",sum);
   *pupilvol=sum;
}
```

# cheng.c

```
/* MODULE TITLE: strehl ratio, R838, mtfa & mtfv calculation
   PROGRAMMER:   Xiaoxue Cheng/ John Handley
   INTENT:       calculate strehl ratio, R838, mtf & mtfv
   ARGUMENTS:
         IN:  re[][], im[][] - pupil function in N*N double array.
         IN:  pupilvol - pupil volume.
        OUT:  strehl, r838, mtfa, mtfv
   RETURNS:      nothing
   CALLS:        fourn
 */

#include<stdio.h>
#include<math.h>
#define T1 0.80                 /* ratio of PSF encircled energy */
#define T2 0.83778487           /* ratio of PSF encircled energy */
#define N 512                   /* array size */
#define WVLENGTH .55            /* 550nm */
#define EYERAD 1500.            /* 1.5 mm */
#define PI (double)4*atan(1)

extern double re[N][N];
extern double im[N][N];
extern double pupilr;

/* Numerical Recipes in C function to compute FFT */
extern void fourn( double *, int *nn, int ndim, int isign);

/* global variable */
double data[N*N*2 + 1];        /* array to hold data for fft */

/* print out mtf raw data(not interpolated ones) from 512x512 array */
void Mtf_raw(double *u, int n)
{
     int i,j;

      printf("mtf@0 degree   90 degree   45 degree   -45 degree\n");
      printf("--------------------------------------------------\n");
      for(i=255; i<n; i++) {
         printf(" %.3f   ", u[n*n/2+i]);
```

```
        printf(" %.3f   ", u[i*n+n/2]);
        printf(" %.3f   ", u[i*n+i]);
        printf(" %.3f\n", u[i*n+n-i]);
    }
}

double volume(int count, double h1, double h2, double h3, double h4)
{
      double v;
      v=(h1+h2+h3+h4)/4.,
      switch(count){
         case 4: break;
         case 1: v=v/4., break;
         case 2: v=v/2.; break;
         case 3: v=3*v/4.;
      }
      return(v);
}

double Mtfa( int n, double *u, double p1, double p2, double max, int ang )
{
   int i, start, end;
   double root2=sqrt(2), area=0.0;

   if(ang==0){
      start=n/2+floor(p1+.5);
      end=n/2+floor(p2+.5);
      for(i=start; i<end; i++)
         area=area+u[n*n/2+i];
   }
   else if(ang==90){
      start=n/2+floor(p1+.5);
      end=n/2+floor(p2 +.5);
      for(i=start; i<end; i++)
         area=area+u[i*n+n/2];
   }
   else if(ang==45){
      start=n/2+floor(p1/root2+.5);
      end=n/2+floor(p2/root2+.5);
      for(i=start; i<end; i++)
         area=area+u[i*n+i];
      area=area*root2;
```

```
    }
    else if(ang==-45){
        start=n/2+floor(p1/root2+.5);
        end=n/2+floor(p2/root2+.5);
        for(i=start; i<end; i++)
            area=area+u[i*n+n-i];
        area=area*root2;
    }
    else
        printf("\nInvalid angle input\n");

    area=area/max;
    return(area);
}

/* find volume of array *u between radius (p1) and (p2)  */
double MtfV( int n, double *u, double p1, double p2, double max )
{
    double dist, v, temp, temp1, sqrt_2=1.415, xc=N/2, yc=N/2;
    int i, j, count=0;

    v = 0.0;
    for( i=0; i<n-1; i++ )
    {
            temp=(i-xc)*(i-xc);
            temp1=(i+1-xc)*(i+1-xc);
            for( j=0; j<n-1; j++)
            {
              dist=sqrt(temp+(j-yc)*(j-yc));
              if( dist >= p2+sqrt_2 || dist <= p1-sqrt_2 )
                    count=0;
              else if( dist <= p2-sqrt_2 && dist >= p1+sqrt_2 )
                    count=4;
              else{
                  count=0;
                  if( dist<p2 && dist>p1 )
                        count=count+1;

                  dist=sqrt(temp+(j+1-yc)*(j+1-yc));
                  if(dist<p2 && dist>p1)
                        count=count+1;
```

```
            dist=sqrt(temp1+(j-yc)*(j-yc));
            if(dist<p2 && dist>p1)
                  count =count+1;

            dist=sqrt(temp1+(j+1-yc)*(j+1-yc));
             if(dist<p2 && dist>p1)
                  count=count+1;
         }
         if (count != 0)
         v=v+volume(count,u[i*n+j],u[(i+1)*n+j],u[i*n+j+1],u[(i+1)*n+j+1]);
      }
   }
   v=v/max;
   return(v);
}

/* find volume of array *u(of size n) within radius r(centroid xc, yc) */
double vol( int n, double *u, double r, double xc, double yc )
{
   double dist, v, temp, temp1;
   int i, j, count=0;

   v = 0.0;
   for( i=0; i<n-1; I++ )
   {
         temp=(i-xc)*(i-xc);
         temp1=(i+1-xc)*(i+1-xc);
         for( j=0; j<n-1; j++)
         {
            dist=sqrt(temp+(j-yc)*(j-yc));
            if(dist >= r+1.415)
                  count=0;
            else if(dist<= r-1.415)
                  count=4;
            else{
                  count=0;
                  if(dist<=r)
                     count=count+1;

                  dist=sqrt(temp+(j+1-yc)*(j+1-yc));
                  if (dist<=r)
                     count=count+1;
```

```
            dist=sqrt(temp1+(j-yc)*(j-yc));
            if(dist<=r)
               count =count+1;

            dist=sqrt(temp1+(j+1-yc)*(j+1-yc));
            if(dist<=r)
               count=count+1;
         }
         if (count != 0)
         v=v+volume(count,u[i*n+j],u[(i+1)*n+j],u[i*n+j+1],u[(i+1)*n+j+1]);
      }
   }
   return(v);
}

void Centroid(int n, double *u, double *xc, double *yc, double *sum)
{
   double xsum=0.0, ysum=0.0, summ=0.0;
   int i, j;

   /* find the centroid of PSF */
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         summ=summ+u[i*n+j];
         xsum=xsum+i*u[i*n+j];
         ysum=ysum+j*u[i*n+j];
      }
   }
   *xc=xsum/summ;
   *yc=ysum/summ;
   fprintf(stderr, "Centroid xc=%.2f yc=%.2f energy:%f\n", *xc, *yc, summ);
   *sum=summ;
}

double binary_search( double start, double end, double percent,int n,
                      double *u, double xc, double yc, double sum )
{
    double val;
    double radius;
    static int counter = 0;
```

```
    radius = (double) (start+end)/2;
    val = vol( n, u, radius, xc, yc )/sum;
    fprintf(stderr,"counter:%d  vol percent: %f radius: %f\n", ++counter, val, radius);
    if ( fabs(val-percent) <.00003 || fabs(start - end) < 0.01 )
        return( radius );

    if( percent < val )
        binary_search( start, radius, percent, n, u, xc, yc, sum);
    else
        binary_search( radius, end, percent, n, u, xc, yc, sum);
}

double FindRadius(int n, double *u, double xc, double yc, double sum, double percent)
{
    double e, r;
    double y;

   /* find radius encircled (percent)% of total energy */

    r = binary_search(0, (double)(n/2), percent, n, u, xc, yc, sum);
    y = vol(n, u, r, xc, yc)/sum;
    e = fabs(y-percent);

    return(r);
}

void Checkboard(int n)
{
    int row, col;

    for( row = 0; row < n; ++row )
      for( col = 0; col < n; ++col )
      {
         if( (row + col)%2 == 0)
         {
            data[2*(row*n + col) + 1] *= -1;
            data[2*(row*n + col) + 2] *= -1;
         }
      }
}

/* print out PSF or MTF image */
```

```
void PrintImg(double max, double min, double *u, FILE *image, char *s)
{
  int col, row;

  if(image == NULL )
  {
    image = fopen(s,"wb");
    if ( image == NULL) exit(1);
    fprintf(stderr,"Writing image %s\n", s);

    fprintf(image,"P5\n");                /* a special format of image */
    fprintf(image,"%d %d\n", N, N);       /* a special format of image */
    fprintf(image,"255\n");               /* maximum grey level of the image */

    for(row=0; row < N; row++)
      for(col=0; col < N; col++)
        putc((unsigned char)255*pow((double)u[row*N+col]/(max-min),0.125 ),
             image);

    fclose(image);
  }
}

void cheng( double pupilvol, double *strehl, double *r838, double *mtfa, double *mtfv )
{
  double xc,yc;
  double sum=0., temp;
  double p1, p2, max, min = 0., area0, area90, area45, area45m, area;
  int   n, i, j, row, col, nn[3], inttemp;
  int   strehl_x, strehl_y;
  double *u, *phase;           /* array to hold 512x512 image */
  FILE *fp, *image = NULL, *image1 = NULL;

  n = N;  /* let n be hard coded to N */

  /* Get raw data, column at a time */
  for(col=0; col < n; col++)
      for(row=0; row < n; row++){
        inttemp=2*(row*n + col);
        data[inttemp + 1] = re[col][row]/pupilvol;
        data[inttemp + 2] = im[col][row]/pupilvol;
      }
```

```
    /* compute forward FFT */
    fprintf(stderr,"Begin FFT.\n");
    nn[0] = 0; nn[1] = N; nn[2] = N;
    Checkboard(n);
    fourn(data, nn, 2, 1);
    fprintf(stderr, "FFT done.\n");

    /* allocate memory for the psf */
    u=(double *)malloc(n*n*sizeof(double));
    if(u == NULL)
       printf("no more dynamic memory to allocate array u.\n");

    /* allocate memory for the phase of OTF */
    phase=(double *)malloc(n*n*sizeof(double));
    if(phase == NULL)
       printf("no more dynamic memory to allocate array phase.\n");

    /* compute power and store in u */
    max = -1.;
    sum = 0.0;
    for(row=0; row < n; row++)
       for(col=0; col < n; col++) {
          inttemp=2*(row*n + col);
          temp = u[row*n+col] = (data[inttemp+1]*data[inttemp+1]
                  + data[inttemp+2]*data[inttemp+2]);
          /* find out maximum of PSF */
          if (temp > max) {
             max = temp;
             strehl_x = row;
             strehl_y = col;
          }
          min = (temp < min) ? temp:min;
          sum+=temp;
       }
    *strehl=max;
    fprintf(stderr,"psf_max: %f pos: %d,%d energy: %f\n", max, strehl_x, strehl_y, sum);
    if( (max-min) <= 0.00000000001) exit(1);

    /* printout PSF image */
/* PrintImg(max, min, u, image, "psf.pgm"); */

    /* print out the central section of PSF */
```

```
/* fprintf(stdout, "max. PSF = %f\n", max);
   fprintf(stdout, "row=%d ---------------------\n", strehl_x);
   for(col=strehl_y; col<n; col++)
      fprintf(stdout, "%f\n", u[strehl_x*n+col]);  */

   Centroid(n, u, &xc, &yc, &sum);

   *r838 =FindRadius( n, u, xc, yc, sum, T2 );


   /* do MTF calculations */
   /* Get raw data, column at a time */
   for(col=0; col < n; col++)
      for(row=0; row < n; row++)
      {
         inttemp=2*(row*n + col);
         data[inttemp + 1] = u[row*n+col];
         data[inttemp + 2] = 0.0;
      }

   /* compute forward FFT */
   fprintf(stderr,"Begin FFT.\n");
   nn[0] = 0; nn[1] = N; nn[2] = N;
   Checkboard(n);
   fourn(data, nn, 2, 1);
   fprintf(stderr, "FFT done.\n");

   /* compute mtf,phase transfer function of OTF and store in u, phase */
   for(row=0; row < n; row++)
      for(col=0; col < n; col++){
         inttemp=2*(row*n + col);
         u[row*n+col] = sqrt ( data[inttemp + 1]*data[inttemp + 1]
                                 + data[inttemp + 2]*data[inttemp + 2]);
   /*     phase[row*n+col] = atan( data[inttemp + 2]/data[inttemp + 1]); */
      }

/* Take one section of phase transfer function */
/*    for (row=0; row<n; row++){
         fprintf(stdout, "%f\n", phase[row*n+n/2]);
         }        */

   max=u[N*N/2+N/2];
```

```
  min=0.0;

/* Mtf_raw(u, n);   */

   /* printout MTF image */
/* PrintImg(max, min, u, image1, "mtf.pgm");*/

   p1=.05*(N/2)*pupilr*2;  /* 0.05 - 0.21 (5-20cyc/deg for cutoff=95cyc/deg) for mtfa */
   p2=.21*p1/.05;
   area0 =  Mtfa(n,u,p1,p2,max,0);
   area90 =  Mtfa(n,u,p1,p2,max,90);
   area45 =  Mtfa(n,u,p1,p2,max,45);
   area45m = Mtfa(n,u,p1,p2,max,-45);
   area=(area0+area90+area45+area45m)/4;  /* average of 0, 90, 45, -45 degree area */

   *mtfa = area;
   *mtfv = MtfV( n, u, p1, p2, max );

   free(u);
   return;
}
```

# fourn.c

```
#undef SWAP
#include <math.h>

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

void fourn(data, nn, ndim, isign)
	double data[];
	int nn[], ndim, isign;
{
	int i1, i2, i3, i2rev, i3rev, ip1, ip2, ip3, ifp1, ifp2;
	int ibit, idim, k1, k2, n, nprev, nrem, ntot;
	double tempi, tempr;
	double theta, wi, wpi, wpr, wr, wtemp;

	ntot=1;
	for (idim=1;idim<=ndim;idim++)
		ntot *= nn[idim];
	nprev=1;
	for (idim=ndim;idim>=1;idim--) {
		n=nn[idim];
		nrem=ntot/(n*nprev);
		ip1=nprev << 1;
		ip2=ip1*n;
		ip3=ip2*nrem;
		i2rev=1;
		for (i2=1;i2<=ip2;i2+=ip1) {
			if (i2 < i2rev) {
				for (i1=i2;i1<=i2+ip1-2;i1+=2) {
					for (i3=i1;i3<=ip3;i3+=ip2) {
						i3rev=i2rev+i3-i2;
						SWAP(data[i3],data[i3rev]);
						SWAP(data[i3+1],data[i3rev+1]);
					}
				}
			}
			ibit=ip2 >> 1;
			while (ibit >= ip1 && i2rev > ibit) {
				i2rev -= ibit;
```

```
                ibit >>= 1;
            }
            i2rev += ibit;
        }
        ifp1=ip1;
        while (ifp1 < ip2) {
            ifp2=ifp1 << 1;
            theta=isign*6.28318530717959/(ifp2/ip1);
            wtemp=sin(0.5*theta);
            wpr = -2.0*wtemp*wtemp;
            wpi=sin(theta);
            wr=1.0;
            wi=0.0;
            for (i3=1;i3<=ifp1;i3+=ip1) {
                for (i1=i3;i1<=i3+ip1-2;i1+=2) {
                    for (i2=i1;i2<=ip3;i2+=ifp2) {
                        k1=i2;
                        k2=k1+ifp1;
                        tempr=wr*data[k2]-wi*data[k2+1];
                        tempi=wr*data[k2+1]+wi*data[k2];
                        data[k2]=data[k1]-tempr;
                        data[k2+1]=data[k1+1]-tempi;
                        data[k1] += tempr;
                        data[k1+1] += tempi;
                    }
                }
                wr=(wtemp=wr)*wpr-wi*wpi+wr;
                wi=wi*wpr+wtemp*wpi+wi;
            }
            ifp1=ifp2;
        }
        nprev *= n;
    }
}
```

# makefile

```
CFILES= wave.c cheng2.c fourn.c
OFILES= wave.o cheng2.o fourn.o
.c.o:
	gcc -c -g $*.c

wave:	${OFILES}
	gcc -o wave ${OFILES} -lm

wave.o: wave.c
cheng2.o: cheng2.c
fourn.o: fourn.c

clean:
	rm -f *.o
	rm -f core
img:
	xloadimage image.pgm&
setenv:
setenv DISPLAY potemkin:0.1
```

## Accuracy test 1.

(1) by the analytical expression for the Airy disk.

The analytical expression for the Airy disk is

$$y = \left[\frac{2J_1(\pi\rho)}{\pi\rho}\right]^2$$

where $\rho$ is the distance from the optical axis in image plane, $y$ the intensity distribution with $\rho$, and $J_1()$ the first-order Bessel function of the first kind.

The 512x512 FFT program output and the corresponding analytical expression values (calculated by Mathcad) for Airy disk are plotted in Fig.A1.1, and some values are listed in Table A1.1 for $0 \le \rho \le 1$. We can see from the table the difference between the two is less than 1% at all points.

Table A1.1 Comparison of Airy Disk Intensity Values between Analytical Expression and the 512x512 FFT Program

| ρ | Analytical expression for Airy disk (y) | 512x512 FFT program output (0.1 of 256 as pupil radius) | Difference |
|---|---|---|---|
| 0 | 1 | 1 | 0.00% |
| 0.1 | 0.9756 | 0.975554 | 0.00% |
| 0.2 | 0.9053 | 0.905178 | 0.01% |
| 0.3 | 0.7975 | 0.797262 | 0.03% |
| 0.4 | 0.6645 | 0.664218 | 0.04% |
| 0.5 | 0.5209 | 0.520467 | 0.08% |
| 0.6 | 0.3806 | 0.380195 | 0.11% |
| 0.7 | 0.2558 | 0.255311 | 0.19% |
| 0.8 | 0.1544 | 0.153977 | 0.27% |
| 0.9 | 0.08027 | 0.07992 | 0.44% |

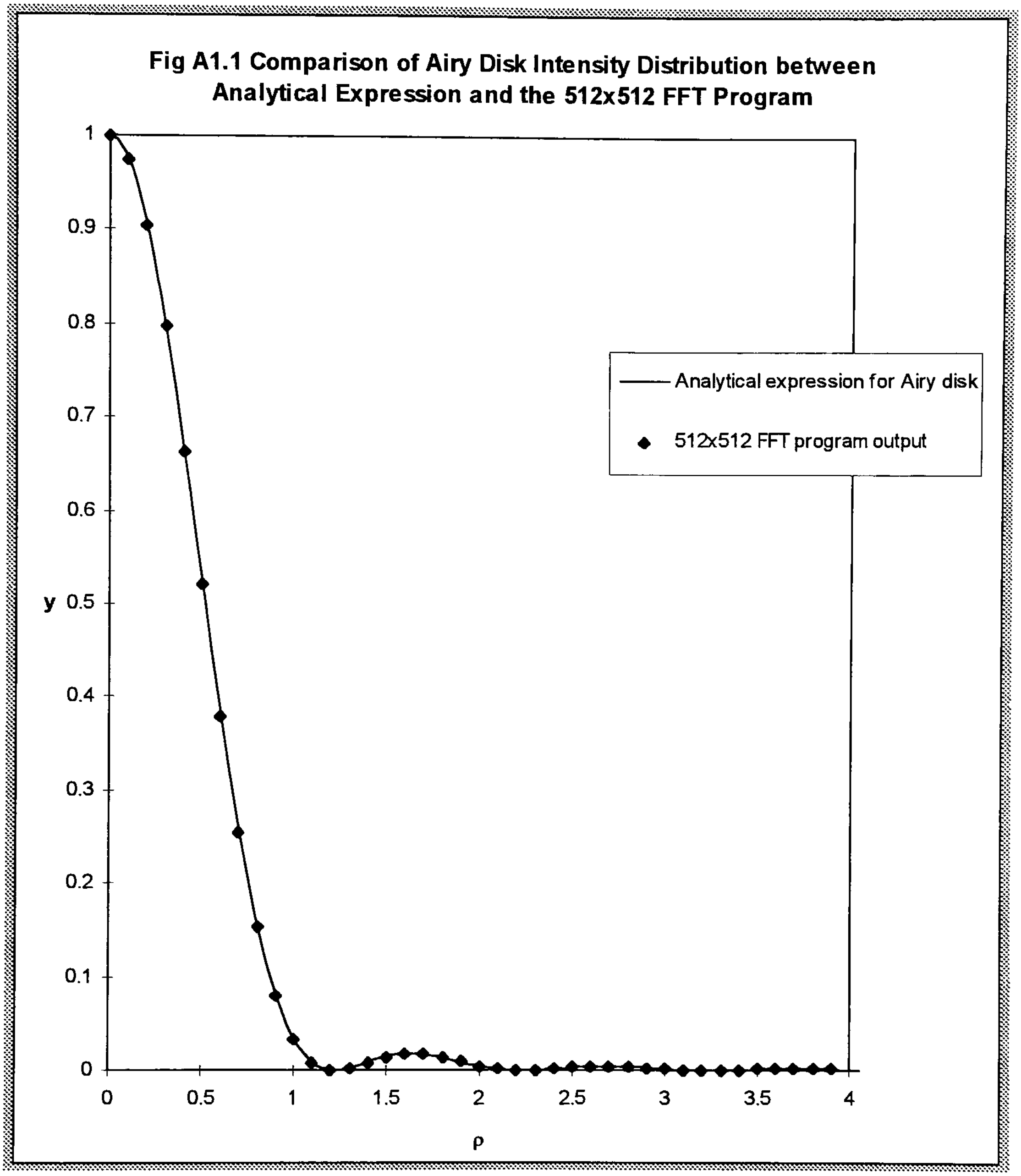


Fig A1.1 Comparison of Airy Disk Intensity Distribution between Analytical Expression and the 512x512 FFT Program
Analytical expression for Airy disk
512x512 FFT program output
y
ρ
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.5
1
1.5
2
2.5
3
3.5
4

## Accuracy test 2.

(2) by the analytical expression for the diffraction-limited MTF.

The analytical expression for the diffraction-limited MTF is

$$D(\sigma) = (1/\pi)( 2\beta - \sigma \sin(\beta) )$$

where $\beta = \arccos(\sigma/2)$ and the value of $\sigma$ is between 0 and 2.

The 512x512 FFT program output and the corresponding analytical expression values (calculated by Excel) for diffraction-limited MTF are plotted in Fig A1.2, and data are listed in Table A1.2 for $0 \le \sigma \le 2$. We can see from the table the difference between the two is less than 1% at most of the points.

Table A1.2 Comparison of Diffraction-limited MTF Values between Analytical Expression and the 512x512 FFT Program

| $\sigma$ | Analytical expression for diffraction-limited MTF | 512x512 FFT program output (0.1 of 256 as pupil radius) | Difference |
|---|---|---|---|
| 0 | 1 | 1 | 0.00% |
| 0.078125 | 0.950276732 | 0.950508283 | 0.02% |
| 0.15625 | 0.900629441 | 0.901016565 | 0.04% |
| 0.234375 | 0.851134457 | 0.851524848 | 0.05% |
| 0.3125 | 0.801868816 | 0.802033131 | 0.02% |
| 0.390625 | 0.752910636 | 0.752541413 | 0.05% |
| 0.46875 | 0.704339523 | 0.704024593 | 0.04% |
| 0.546875 | 0.656237002 | 0.656474806 | 0.04% |
| 0.625 | 0.608687012 | 0.608925019 | 0.04% |
| 0.703125 | 0.56177646 | 0.562350129 | 0.10% |
| 0.78125 | 0.515595863 | 0.516734411 | 0.22% |
| 0.859375 | 0.470240113 | 0.471126556 | 0.19% |
| 0.9375 | 0.425809393 | 0.426493596 | 0.16% |
| 1.015625 | 0.382410295 | 0.382819809 | 0.11% |
| 1.09375 | 0.340157226 | 0.340128781 | 0.01% |
| 1.171875 | 0.299174204 | 0.299371821 | 0.07% |
| 1.25 | 0.259597198 | 0.26054893 | 0.37% |
| 1.328125 | 0.221577288 | 0.222708797 | 0.51% |
| 1.40625 | 0.185285032 | 0.185827836 | 0.29% |
| 1.484375 | 0.150916769 | 0.150896669 | 0.01% |
| 1.5625 | 0.118704117 | 0.118874466 | 0.14% |
| 1.640625 | 0.088929183 | 0.089761229 | 0.94% |
| 1.71875 | 0.061950941 | 0.062589922 | 1.03% |
| 1.796875 | 0.038256448 | 0.038327581 | 0.19% |
| 1.875 | 0.018579754 | 0.018923997 | 1.85% |
| 1.953125 | 0.004292082 | 0.005338344 | 24.38% |

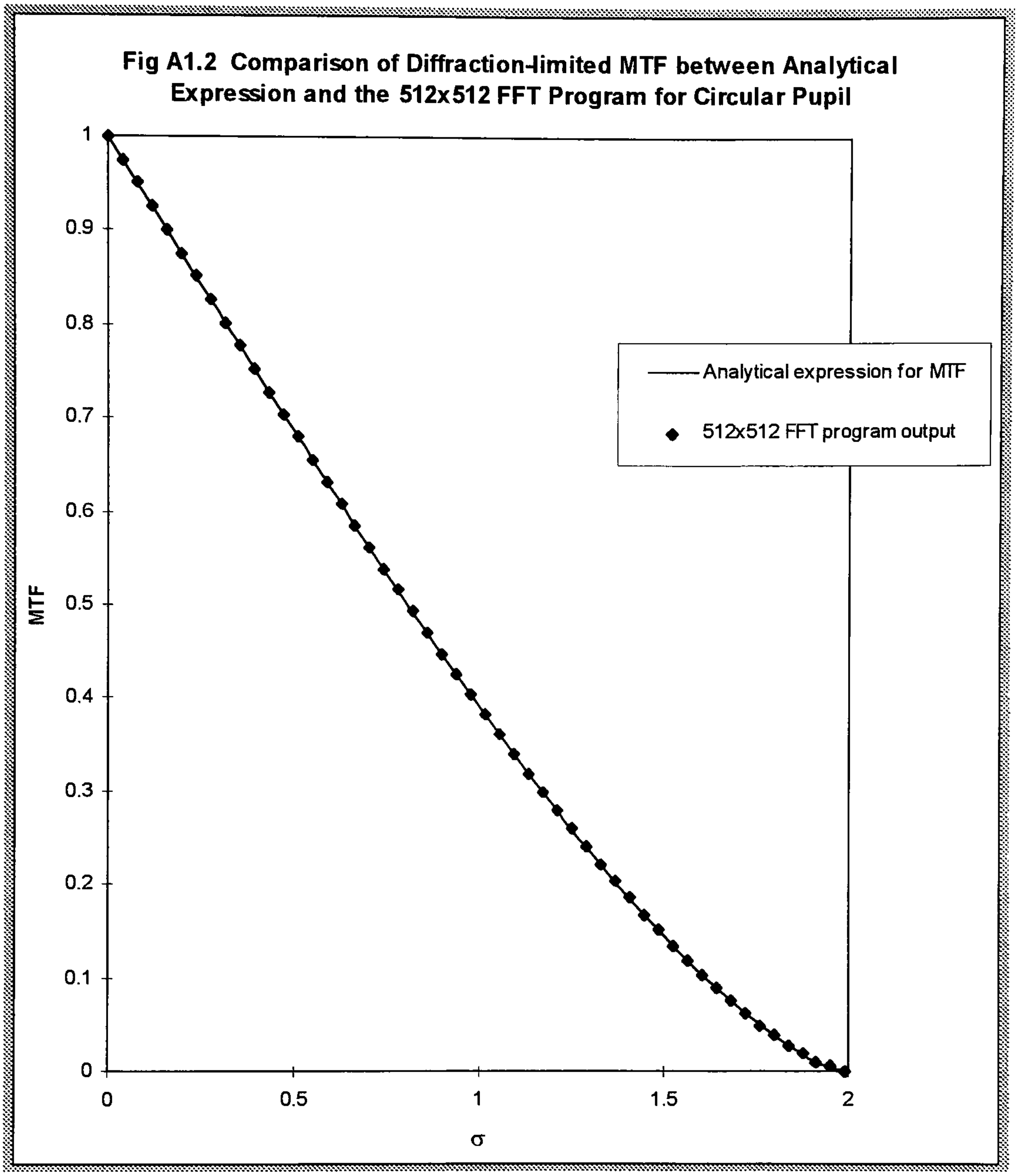


Fig A1.2 Comparison of Diffraction-limited MTF between Analytical Expression and the 512x512 FFT Program for Circular Pupil
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
MTF
0
0.5
1
1.5
2
σ
Analytical expression for MTF
512x512 FFT program output

## Accuracy test 3.

(3) by the analytical expression for the encircled energy of the Airy disk.

The analytical expression for the encircled energy of the Airy disk is[15]

$$\text{Encircled Energy } (\rho) = 1 - J_0^2(\pi\rho) - J_1^2(\pi\rho)$$

where $\rho$ is the distance from the optical axis in image plane, $J_0()$ and $J_1()$ are the 0th and 1st-order Bessel function of the first kind.

The analytical expression gives about 83.8% (83.778487%, more accurately) encircled energy for Airy disk at $\rho = 1.22$ which is the radius of the 1st dark ring. The 512x512 FFT program gives the same encircled energy for Airy disk at $\rho = 1.21875$, which differs from 1.22 about 0.1%.

## Accuracy test 4.

(4) by the analytical expression for the MTF in the presence of defocus.

The analytical expression for the MTF in the presence of defocus is[16]

$$D(\sigma)=\frac{4}{\pi a}\cos\frac{a\sigma}{2}\left\{\beta J_1(a)+\frac{1}{2}\sin(2\beta)(J_1(a)-J_3(a))-\frac{1}{4}\sin(4\beta)(J_3(a)-J_5(a))+\ldots\right\}$$

$$-\frac{4}{\pi a}\sin\frac{a\sigma}{2}\left\{\sin(\beta)(J_0(a)-J_2(a))-\frac{1}{3}\sin(3\beta)(J_2(a)-J_4(a))+\frac{1}{5}\sin(5\beta)(J_4(a)-J_6(a))-\ldots\right\}$$

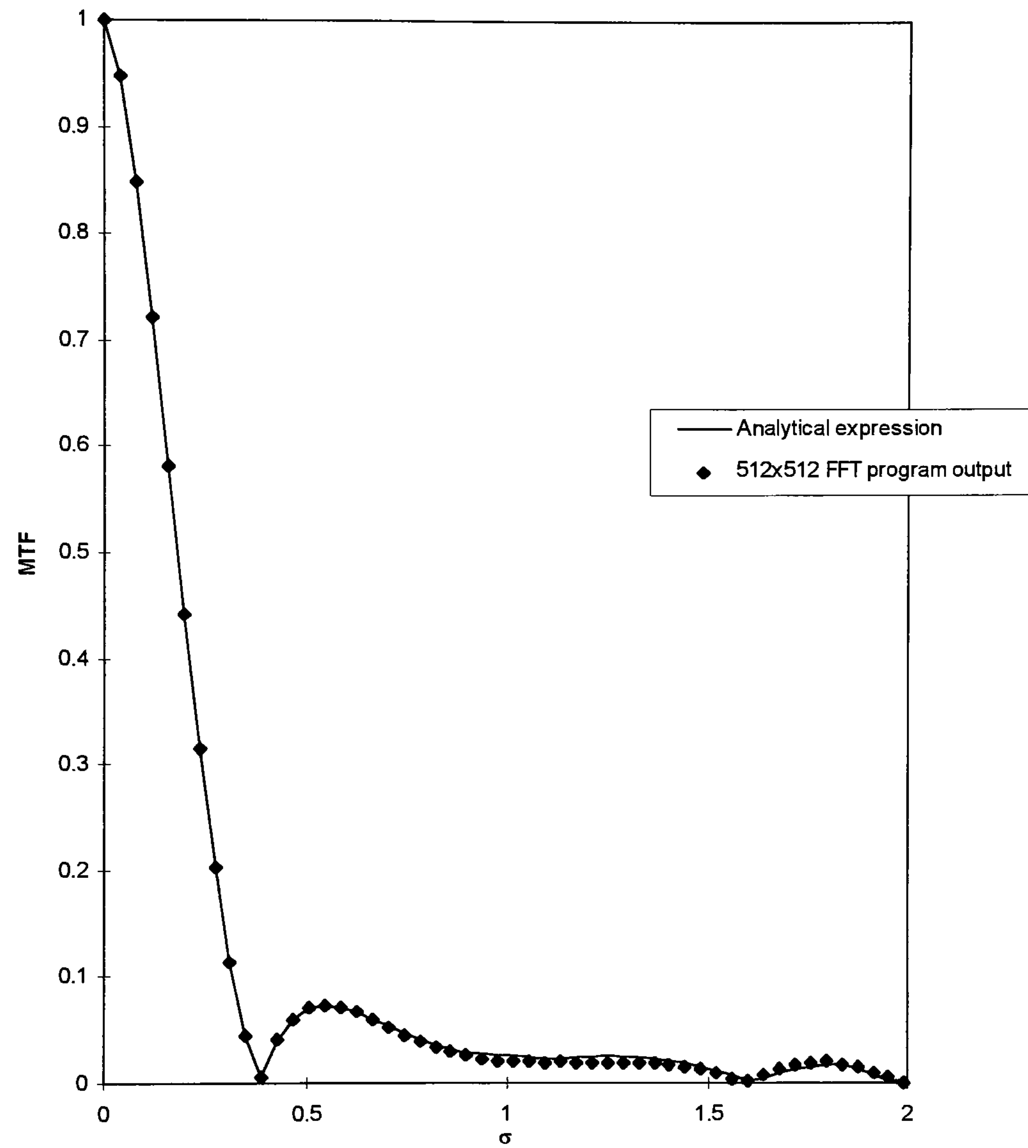
where $a=\frac{4\pi}{\lambda}W_{20}\sigma, \beta=\arccos\frac{\sigma}{2}$

The 512x512 FFT program output and the corresponding analytical expression values (calculated by Excel, to the order of 7th and 8th Bessel functions) for the MTF in the presence of defocus ($W_{20} = 1\lambda$) are plotted in Fig A1.3, and data are listed in Table A1.3 for $0 \le \sigma \le 2$. We can see from the table the difference between the two is less than 1% at the significant points.

Table A1.3 Comparison of MTF Values for Defocus $W_{20}=1\lambda$ between Analytical Expression and the 512x512 FFT Program

| $\sigma$ | Analytical expression $D(\sigma)$ | 512x512 FFT program output (0.1 of 256 as pupil radius) | Difference(%) |
|---|---|---|---|
| 0 | 1 | 1 | 0.00% |
| 0.039063 | 0.947236 | 0.947324145 | 0.01% |
| 0.078125 | 0.848879 | 0.849001124 | 0.01% |
| 0.117188 | 0.721613 | 0.721722107 | 0.02% |
| 0.15625 | 0.581658 | 0.581714403 | 0.01% |
| 0.195313 | 0.4427 | 0.442673732 | 0.01% |
| 0.234375 | 0.314833 | 0.314734301 | 0.03% |
| 0.273438 | 0.204362 | 0.204201489 | 0.08% |
| 0.3125 | 0.114205 | 0.114015708 | 0.17% |
| 0.351563 | 0.044638 | 0.044428546 | 0.47% |

**Fig A1.3 Comparison of MTF between analytical expression and the 512x512 FFT program for defocus $W_{20}=1\lambda$**



## Accuracy test 5.

(5) by the encircled energy function for rotationally symmetric aberrations, which was computed independently in polar coordinates by integrating along the radius.

The PSF for rotationally symmetric aberrations is[17]

$$G^2(\rho) = \left| \; 2\pi \int_0^1 e^{ikW(r)} J_0(2\pi\rho r) r dr \; \right|^2$$

where G(ρ) is the Fourier transform of the entrance pupil function for optical system, W(r) the wavefront aberration, $J_0()$ the 0th-order Bessel function of the first kind, r the pupil radius, ρ the image field radius and k = 2π/λ.

Then the encircled energy can be computed by integrating PSF along the radius ρ:

$$E(\rho') = \int_0^{2\pi} \int_0^{\rho'} G^2(\rho) \rho d\rho d\theta = K \int_0^{\rho'} \rho \left| \int_0^1 e^{ikW(r)} J_0(2\pi\rho r) r dr \right|^2 d\rho$$

where K is a constant. The above integration in polar coordinate can be evaluated by Mathcad. The comparison for $R_{84}$ between the Mathcad results and 512x512 FFT program output is shown in Table A1.4 for different rotationally symmetric aberrations. The agreement between the two is better than 1%.

Table A1.4 Comparison of $R_{84}$ (normalized by that of Airy Disk) between Mathcad and the 512x512 FFT program for rotationally symmetric aberrations

| Rotationally symmetric aberrations | $R_{84}$ by Mathcad | $R_{84}$ by 512x512 FFT program | Difference |
|---|---|---|---|
| $W_{20} = 0.5l$ | 2.028 | 2.029 | 0.05% |
| $W_{40} = 0.5l$ | 2.474 | 2.482 | 0.32% |
| $W_{60} = 0.5l$ | 2.813 | 2.821 | 0.28% |
| $W_{80} = 0.5l$ | 2.921 | 2.937 | 0.55% |
| $W_{20} = 1l$ | 3.237 | 3.247 | 0.31% |
| $W_{40} = 1l$ | 4.597 | 4.603 | 0.13% |
| $W_{60} = 1l$ | 5.495 | 5.512 | 0.31% |
| $W_{80} = 1l$ | 6.135 | 6.151 | 0.26% |

# Appendix B.

## DOG target generation program

The DOG target generation program named *dog.c* is to generate a rotationally symmetric difference-of-gaussians (DOG) luminance profile on a high resolution 19" monitor which is connected to a personal computer installed with a *Pepper Pro-1280 display board* from Number Nine Computer Systems. This program was developed from a sinusoids generation program written by T. Kim.

The program starts by providing users an interface screen with several options. There are three meaningful options ("*R*", "*C*", "*E*") used in the DOG experiment. The option "*R*" allows to change the DOG radius; "*C*" allows to change the DOG contrast; "*E*" allows to conduct the random double staircase experiment and record the contrast settings with initial high and low starting contrast determined by the program. In the DOG experiment, first, one needs to select "*R*" to enter the correct DOG radius which should result in the desired spatial frequency range for the DOG target on the monitor viewed from certain distance, then one can select "*E*" to conduct the random double staircase experiment.

# dog.c

```
/*
Name:   DOG GENERATOR modified from SINUSOID GENERATOR
*/

#include <nnios.h>              /* Required for all NNIOS programs */
#include <nnioslib.h>           /* Language binding header file    */
#include <math.h>
#include <conio.h>
#include <process.h>
#include <stdio.h>
#include <graph.h>
#define CLEAR   "\x1B[2J"
#define NORMAL  "\x1B[0m"
#define BOLD    "\x1B[1m"
#define BLINK   "\x1B[5m"
#define REVERSE "\x1B[7m"
#define ERASE   "\x1B[K"
#define CYAN    3
#define MAGENTA 5
#define YELLOW  14
#define pi      3.141592653
#define pix_x   0.296875
#define pix_y   0.2734375
#define half    0.50
#define yrange  70
#define SPKRON  3

DEV      device;                /* Info about the current device   */
BITMAP   bitmap;                /* Data for the app's one bitmap   */
WINDOW   window;                /* And for the app's one window    */
DEVCON   dev;                   /* Current device configuration    */
WORD     devCount;
HANDLE   font;                  /* Handle of loaded text font      */
POINT    strSize;               /* Size of a text string in pixels */
POINT    point[2];
BYTE     message[64];           /* Buffer for text message         */
COLOR    palette[256];

double k12gamma[3][3] = { {  1.001,  0.0,  2.202 },
```

```
                                { 1.007, 0.0, 2.744 },
                                { 1.006, 0.0, 2.430 } };

double XYZtoRGB[3][3] = { {  0.116353101, -0.031400862, -0.037417107 },
                                { -0.012552165,  0.064952516, -0.020056448 },
                                {  0.010176622,  0.014034621,  0.001641835 } };

float freq_array[6] = { 10.74, 0.0, 0.0, 0.0};

float angle_array[4] = {0.0, 4.0, 8.0};

void ShowSet(float, float, float, float);
void ShowPattern(float, int, int);
int  LUT_RGB(float, int[]);
static int round(double);
void beep(int);
float increase(float,float);
float decrease(float,float);
float convert(float);

int    b_average=128;

main (int argc, char *argv[])
{
  WORD    x1, x2;
  int     b_average=128;
  int     x, color, i, j, g1, g2, tf, j0, j1;
  int     dc[3];

  float   radius=1.0;
  float   dist = 3.0;
  float   c1 = 1.0;
  float   logc1, logc2;
  float   ofreq, logc12, c12;
  float   logc_incre = 0.1;
  char    ch;
  FILE    *fptr1;

  int     c1x1, c1y1, c1x2, c1y2, c2x1, c2y1, c2x2, c2y2;
  int     choice, oage;
  char    osex;
  char    oname[20], buffer[80];
```

```
float   oangle, c_incre;

if (CheckNNIOS ())                /* Try to initialize NNIOS          */
 {
  printf ("This application requires the NNIOS device driver.\n\r");
  exit (-1);
  }

if ((devCount = DeviceCount ()) == 0)
  {
  printf ("No Pepper Graphics System products are installed.\n\r");
  exit (-1);
  }

DeviceInfo (&device, devCount);

OpenCommChannel (&device, "NNIOS Program");
GetDeviceConfig (&dev);
if (dev.deviceClass == 40 | dev.deviceClass == 41 )
  SetDeviceConfig (40);
else
  SetDeviceConfig (50);

GetDeviceConfig (&dev);

BitCreate (&bitmap, dev.displaySize.x, dev.displaySize.y, dev.maxDepth,
          FALSE, 0, "Template");

for (color=0; color<256; color++)
  {
  tf=LUT_RGB((color/255.0), dc);
  if (tf==0) for (i=0; i<3; i++) dc[i]=0;
  palette[color].r=dc[0];
  palette[color].g=dc[1];
  palette[color].b=dc[2];
  SetLUT(0,TRUE,(CINDEX)color,(CINDEX)1,&palette[color]);
  }

BMClear (bitmap.handle, (CINDEX)b_average);

WinCreate (&window, bitmap.handle, 0, 0, dev.displaySize.x,
          dev.displaySize.y, 0, 0, dev.maxDepth, 0, 0, 0);
```

```
/* Set the default values */

  g1 = 128 - 128*c1;
  g2 = 128 + 128*c1;
  if (g2>255) g2=255;

  ofreq = 10.74;

  ShowPattern(radius, g1, g2);

/* Control Screen Block */

  ShowSet(radius, c1, ofreq, dist);
  while (1)
    {
    printf("\x1B[22;55f");
    ch = getch();
    switch (ch)
        {
        case 'S':
        case 's':
          printf(CLEAR);
          printf("\x1B[4;5fbitmap");
          printf("\x1B[6;10fhandle          %5d", bitmap.handle);
          printf("\x1B[7;10fsize.origin.x    %5d size.extents.x   %5d",
                            bitmap.size.origin.x, bitmap.size.extents.x);
          printf("\x1B[8;10fsize.origin.y    %5d size.extents.y   %5d",
                            bitmap.size.origin.y, bitmap.size.extents.y);
          printf("\x1B[9;10ftype            %5d", bitmap.type);
          printf("\x1B[10;10fdepth           %5d    lock %5d", bitmap.depth,
                            bitmap.lock);
          printf("\x1B[13;5fwindow");
          printf("\x1B[15;10fwindowHandle    %5d    bitmapHandle    %5d",
                            window.windowHandle, window.bitmapHandle);
          printf("\x1B[16;10fdisplay.origin.x %5d display.extents.x  %5d",
                            window.display.origin.x, window.display.extents.x);
          printf("\x1B[17;10fdisplay.origin.y %5d display.extents.y  %5d",
                            window.display.origin.y, window.display.extents.y);
          printf("\x1B[18;10fbitmapOrigin.x   %5d bitmapOrigin.y    %5d",
                            window.bitmapOrigin.x, window.bitmapOrigin.y);
          printf("\x1B[19;10fdepth           %5d    priority    %5d", window.depth,
                            window.priority);
```

```
printf("\x1B[20;10fattributes      %5d    zoomSlot      %5d",
                    window.attributes, window.zoomSlot);
  printf("\x1B[24;10fHit any key");
  getch();
  ShowSet(radius, c1, ofreq, dist);
  printf("\x1B[22;53f");
  break;

case 'R':
case 'r':
  do
  {
    printf("\x1B[24;25fEnter radius [cm]:");
    printf("\x1B[24;44f");
    scanf("%f", &radius);
    printf("\x1B[24;1f%s", ERASE);
  }
  while(radius<0);
  printf("\x1B[9;45f[%5.2fcm]", radius);

  printf("\x1B[15;45f[ %5.1fcyc/deg]", ofreq);

  ShowPattern(radius, g1, g2);
  break;

case 'C':
case 'c':
  do
  {
    printf("\x1B[24;25fEnter contrast range [ 0 - 1 ]:");
    printf("\x1B[24;59f");
    scanf("%f", &c1);
    printf("\x1B[24;1f%s", ERASE);
  }
  while((c1<0)||(c1>1));
       printf("\x1B[11;45f[ %7.4f ]", c1);

       c1=c1/(1-.40845*c1);  /* for 128 background illu. */

  g1 = 128 - 128*c1;
  g2 = 128 + 128*c1;
  if (g2>255) g2=255;
```

```
ShowPattern(radius, g1, g2);
   break;

 case 'F':
 case 'f':
   do
   {
     printf("\x1B[24;25fEnter frequency :");
     printf("\x1B[24;45f");
     scanf("%f", &freq);
     printf("\x1B[24;1f%s", ERASE);
   }
   while(freq<=0);
   printf("\x1B[13;45f[ %5.1f ]", freq);
   ofreq = freq*pi/radius*dist*100.0/360.0;
   printf("\x1B[15;45f[ %5.1fcyc/deg]", ofreq);
        ShowPattern(radius, g1, g2);
   break;

 case 'O':
 case 'o':
   do
   {
     printf("\x1B[24;25fEnter frequency :");
     printf("\x1B[24;45f");
     scanf("%f", &ofreq);
     printf("\x1B[24;1f%s", ERASE);
   }
   while(ofreq<=0);
   printf("\x1B[15;45f[ %5.1fcyc/deg]", ofreq);


   ShowPattern(radius, g1, g2);
   break;

 case 'D':
 case 'd':
   do
   {
     printf("\x1B[24;25fEnter viewing distance :");
     printf("\x1B[24;50f");
     scanf("%f", &dist);
```

```
            printf("\x1B[24;1f%s", ERASE);
          }
          while(dist<=0);
          printf("\x1B[17;45f[ %5.1fm]", dist);
/*        ofreq = freq*pi/radius*dist*100.0/360.0;
               printf("\x1B[15;45f[ %5.1fcyc/deg]", ofreq);          */
          break;

        case 'E':
        case 'e':
          if ( (fptr1 = fopen(argv[1], "a")) ==NULL) {
               WinDiscard (window.windowHandle);
               BitDiscard (bitmap.handle);
               CloseCommChannel ();
               printf(CLEAR);
               printf ("can not find date file\n");
               exit(-1);
          }
          printf("\x1B[24;25fEnter observer's name :");
          printf("\x1B[24;50f");
          gets(oname);
          printf("\x1B[24;1f%s", ERASE);
          printf("\x1B[24;25fEnter observer's age :");
          printf("\x1B[24;50f");
          scanf("%d", &oage);
          fflush(stdin);
          printf("\x1B[24;1f%s", ERASE);
          printf("\x1B[24;25fEnter observer's sex :");
          printf("\x1B[24;50f");
          scanf("%c", &osex);
          printf("\x1B[24;1f%s", ERASE);

          _setvideomode(_ERESCOLOR);
          _clearscreen(_GCLEARSCREEN);
          _setviewport(0, 0, 640, 209);
          _settextwindow(16, 1, 25, 80);
          _setbkcolor(4L);
          _clearscreen(_GWINDOW);
          _settextcolor(YELLOW);
          fprintf(fptr1, "name : %s\n", oname);
          fprintf(fptr1, " age : %d\n", oage);
          fprintf(fptr1, " sex : %c\n", osex);
```

```
_settextposition(1,1);
sprintf(buffer, "STATUS observer: %s\n", oname);
_outtext(buffer);
_settextposition(2,3);
sprintf(buffer, "pattern radius : %5.2f\n", radius);
_outtext(buffer);
for (i=0; i<3; i++)  /* for different angle */
{
  oangle = angle_array[i];
  _settextposition(10, 5);
  sprintf(buffer, "Set the telescope angle at %4.1f degree, and hit the Enter",
                  oangle);
  _outtext(buffer);
  getch();
  _settextposition(10, 5);
  sprintf(buffer, " ");
  _outtext(buffer);
  _settextposition(3,3);
  sprintf(buffer, "viewing angle : %4.1f degree", oangle);
  _outtext(buffer);
       fprintf(fptr1, "\n\n angle   %4.1f\n", oangle);

  for (j=0; j<3; j++)  /* for different frequency */
  {
      j0=0;
      j1=0;
      ofreq=freq_array[j];
      if (ofreq==25.0) logc1=0.0;
      else logc1 = 0.3; /*starting point: corresponds to contrast .5 or so*/
      logc2 = 1.5;  /*staring point: corr. to contrast .03 or so */
      _clearscreen(_GVIEWPORT);
      fprintf(fptr1, " frequency : %6.2f cyc/deg\n", ofreq);
      _settextposition(4,3);
      sprintf(buffer, "observing frequency : %6.2f cyc/deg", ofreq);
      _outtext(buffer);
/*    freq= ofreq/pi*radius/dist/100.0*360.0;           */
      c1x1 = 20;
      c1y1 = (-logc1+3.0)*yrange;
      c2x1 = 20;
      c2y1 = (-logc2+3.0)*yrange;

      fprintf(fptr1, "choice  log(contrast sensitivity) Contrast\n");
```

```
do
{
         float real_contrast;
         int max, min;
         long int l;
        if (rand()<16384) choice=1;
        else choice=0;
         if (choice)
        {
          logc12 = logc1;
          j0++;
        }
        else
        {
          logc12 = logc2;
          j1++;
        }

         _settextposition(5,3);
         sprintf(buffer, "log(contrast sensitivity): %4.3f  j0:%2d  j1:%2d",
                                   logc12, j0, j1);
         _outtext(buffer);

         c12= 1/pow(10, logc12);

         c12=c12/(1-.40845*c12);  /* for 128 background illu. */

        g1 = 128 - 128*c12;
        g2 = 128 + 128*c12;
         if (g2>255) g2=255;

        max = (int)(128+(g2-g1)*.704);
                /* .704 is the normalized range of DOG above zero */
        min = (int)(128-(g2-g1)*.296);
                /* .296 is the normalized range of DOG below zero */

        real_contrast=(float)(max-min)/(float)(max+min);

        fprintf(fptr1,"%1d  %10.4f %27.6f\n", choice,
                      log10(1/real_contrast), real_contrast);

        BMClear (bitmap.handle, (CINDEX)b_average);
```

```
        ShowPattern(radius, g1, g2);
        beep(300);
        _settextposition(10, 5);
        sprintf(buffer, "         ");
        _outtext(buffer);
         for(l=0;l<1.2e5;l++) {
                double a,b;
                a=(double) (l+i);
                b=(double) (l*i);
                a=b*b;
        }
         beep(5e3);


    _settextposition(10, 5);
    sprintf(buffer, "Enter y, n, q     ");
    _outtext(buffer);

    ch=getch();
    if (ch=='q') break;
    else if ((ch!='y')&&(ch!='n'))
        {
        _settextposition(10, 5);
        sprintf(buffer, "Enter either 'y' or 'n'      ");
        _outtext(buffer);
        ch=getch();
        _settextposition(10, 5);
        sprintf(buffer, "                             ");
        _outtext(buffer);
        }
    else if (ch=='y')
        {
        if (choice)
          {
          logc1= increase(logc1,logc_incre);
          c1x2 = c1x1 + 10;
          c1y2 = (-logc1+3.0)*yrange;
          _setcolor(MAGENTA);
          _moveto(c1x1, c1y1);
          _lineto(c1x2, c1y2);
          c1x1 = c1x2;
          c1y1 = c1y2;
```

```
          }
        else
          {
          logc2 = increase(logc2,logc_incre);
          c2x2 = c2x1 + 10;
          c2y2 = (-logc2+3.0)*yrange;
          _setcolor(CYAN);
          _moveto(c2x1, c2y1);
          _lineto(c2x2, c2y2);
          c2x1 = c2x2;
          c2y1 = c2y2;
          }
        }
    else
        {
        if (choice)
          {
          logc1= decrease(logc1,logc_incre);
          c1x2 = c1x1 + 10;
          c1y2 = (-logc1+3.0)*yrange;
          _setcolor(MAGENTA);
          _moveto(c1x1, c1y1);
          _lineto(c1x2, c1y2);
          c1x1 = c1x2;
          c1y1 = c1y2;
          }
        else
          {
          logc2 = decrease(logc2 , logc_incre);
          c2x2 = c2x1 + 10;
          c2y2 = (-logc2+3.0)*yrange;
          _setcolor(CYAN);
          _moveto(c2x1, c2y1);
          _lineto(c2x2, c2y2);
          c2x1 = c2x2;
          c2y1 = c2y2;
          }
        }
        if ((j0==55)||(j1==55)) beep(300);
}
while((j0<60)||(j1<60));
_settextposition(10, 5);
```

```
                sprintf(buffer, "Continue different frequency ? ");
                _outtext(buffer);
                if (getch()=='n') break;

            }
            _settextposition(10, 5);
            sprintf(buffer, "Continue different angle ?        ");
            _outtext(buffer);
            if (getch()=='n') break;

          }
          fclose(fptr1);
          _setvideomode(_DEFAULTMODE);

        case 'Q':
        case 'q':

          WinDiscard (window.windowHandle);
          BitDiscard (bitmap.handle);
          CloseCommChannel ();
          printf(CLEAR);
               exit(-1);

        default:
          printf("\x1B[24;25fEnter again");
        }
    }
}


void ShowSet(float radius, float c1, float ofreq, float dist)
{
  printf(CLEAR);
  printf("\x1B[31m%s\x1B[2;26fNumber Nine PepperPro 1280", BOLD);
  printf("\x1B[37m\x1B[4;31fDOG GENERATOR%s", NORMAL);
  printf("\x1B[34m%s\x1B[7;20fS%show settings", BOLD, NORMAL);
  printf("\x1B[34m%s\x1B[9;20fR%sadius of pattern", BOLD, NORMAL);
  printf("\x1B[9;45f[%5.2fcm]", radius);
  printf("\x1B[34m%s\x1B[11;20fC%sontrast Range", BOLD, NORMAL);
  printf("\x1B[11;45f[ %7.4f ]", c1);
/*  printf("\x1B[34m%s\x1B[13;20fF%srequency on the screen", BOLD, NORMAL);
  printf("\x1B[13;45f[ %5.1f ]", freq);
```

```
  printf("\x1B[34m%s\x1B[15;20fO%sbserving frequency", BOLD, NORMAL);
  printf("\x1B[15;45f[ %5.1fcyc/deg]", ofreq);    */
  printf("\x1B[17;20fObserving r\x1B[34m%se%scord", BOLD, NORMAL);
/*  printf("\x1B[17;45f[ %5.1fm]", dist);        */
  printf("\x1B[34m%s\x1B[19;20fQ%suit", BOLD, NORMAL);
  printf("\x1B[22;20fEnter choices [s,r,c,e,q] ·");
}


void ShowPattern(float radius, int g1, int g2)  /* written by cheng */
{
  int    x, y, color, x_radius, y_radius, xx;
  int    g2_g1;
  float  radius_square, y_pix, powx, powy;
  long int integrat=0, pixelcount=0;
  float  averageilluminance=0.0;
  char   ch;
  float  dogedge=0.0, x_cordinate=0.0, y_cordinate=0.0;

  g2_g1 = g2 - g1;
  printf("max = %3d, min = %3d\n",(int) (128+g2_g1*.704),
                      (int)(128-g2_g1*.296) );

  x_radius=radius*10/pix_x;   /* pix_x(y): pixel width in x(or y) axis */
  y_radius=radius*10/pix_y;   /* radius along y_axis with unit in pixel */

  radius_square=radius*radius*100;  /* radius squared in mm */
  y_pix = pix_y*pix_y;    /* distance squared per pixel in y direction */

  dogedge = (3*exp(-64)-2*exp(-64/2.25))/1.42*g2_g1 + 128;
 /* last number 128 is for background illu.       */

  b_average = dogedge;  /*let background has same illuminance as DOG edge*/
 /* printf("edge & background illuminance = %f\n", dogedge);    */

  BMClear (bitmap.handle, (CINDEX)b_average);

  for (y=0; y<=y_radius; y++)
  {
        y_cordinate = y*pix_y*.1/radius;  /* radius normalized to 1 */
        powy = y_cordinate * y_cordinate;
        x=sqrt(radius_square - y_pix*y*y)/pix_x;
```

```
        for(xx=-x; xx<=0; xx++){
            x_cordinate = xx*pix_x*.1/radius;  /* radius normalized to 1 */
            powx = x_cordinate * x_cordinate;

            color=(3*exp(-(powx+powy)*64)-
                        2*exp(-(powx+powy)*64/2.25))/1.42*g2_g1 + 128;
             /* 1.42 is the whole range for DOG */
             /* last number 128 is for background illu.      */

          /* integrat += color;
            pixelcount++;              */

            SetColor (bitmap.handle, (CINDEX)color, (CINDEX)0);
            point[0].y = point[1].y = 512+y;
            point[0].x = point[1].x = 640+xx;
            PolyLine (bitmap.handle, 2, point);

            point[0].x = point[1].x = 640-xx;
            PolyLine (bitmap.handle, 2, point);

            point[0].y = point[1].y = 512-y;
            PolyLine (bitmap.handle, 2, point);

            point[0].x = point[1].x = 640+xx;
            PolyLine (bitmap.handle, 2, point);

        }
    }

 /* averageilluminance=(float)integrat/(float)pixelcount;
  printf("average illuminance = %f\n", averageilluminance);
  scanf("%c", &ch);                                        */
}

int LUT_RGB(float cons, int dac[])
{

  int   i,j;
  double x, y, X, Y, Z;
  double XYZ[3];
  double RGB[3];
```

```
  x = 0.2824;        /* Assumed white points */
  y = 0.3027;
  Y = cons * 38.78;  /* normalized contrast multiply max Y */

  /* Transform x,y,Y to X,Y,Z */

  if (Y==0)
    return(0);
  else
    {
    XYZ[0] = Y*x/y;
    XYZ[1] = Y;
    XYZ[2] = (1.0 - x - y)/y*Y;
    }

  /* Transform X,Y,Z to R,G,B */

  for (i=0; i<3; i++)
    {
    RGB[i] = 0;
    for (j=0; j<3; j++) RGB[i] += XYZtoRGB[i][j]*XYZ[j];
    }

  /* Transform R,G,B to DCR, DCG, DCB */

  for (i=0; i<3; i++)
    {
    dac[i] = 0;
    if (RGB[i]<0) return(0);
    dac[i] = round(255*(pow(RGB[i], (1/k12gamma[i][2])) -k12gamma[i][1])
                      /k12gamma[i][0]);
    }

  return(1);
}
static int round(double value)
{
  double intpart;

  value = modf(value, &intpart);

  if (value<half) return((int)intpart);
```

```
  else return((int)intpart+1);
}

void beep(int l)
{
  unsigned status, part_ratio;
  double a, b, c, d;
  int   i;

  status = inp (0x61);
  outp(0x61, (status | SPKRON));
  for(i=0; i<l; i++)
  {
    a = (double)(i-1);
    b = (double)(i+1);
    c = (double)(i*i);•
    d = a*b - c;
  }
  status = inp (0x61);
  outp(0x61, (status & ~SPKRON));
}


float increase(float log_contrast, float step)
{       char buffer[80];
        float tmp;

        tmp=log_contrast;
        log_contrast=log_contrast+step;
        while ( (log_contrast-tmp) == 0.0 ){
                if (log_contrast>2.101) {
                        _settextposition(10,5);
                        sprintf(buffer,"WARNING: contrast is close to zero");
                        _outtext(buffer);
                        beep(300);
                        return(tmp);
                }
                log_contrast=log_contrast+step;
        }
        return (log_contrast);
}
float decrease(float log_contrast, float step)
```

```
{       char buffer[80];

        if (log_contrast-step<(-0.0) ) {
                _settextposition(10,5);
                sprintf(buffer,"WARNING: contrast is close to 1");
                _outtext(buffer);
                beep(300);
                return(0.0);
        }
        log_contrast=log_contrast-step;
        while ( step == 0.0 ){
                log_contrast=log_contrast-step;
        }
        return (log_contrast);
}

float convert(float log_contrast)       /* not used in main by cheng */
{
        float contrast,real_contrast,real_log_contrast;
        int     maximum,minimum;
        contrast=1/pow(10,log_contrast);
        maximum=128+128*contrast;
        minimum=128-128*contrast;
        printf("--->%3d %3d\n", maximum, minimum);
        if (maximum>255) maximum=255;

        real_contrast=(double)(maximum-minimum)/(double)(maximum+minimum);
        real_log_contrast= -log10 (real_contrast);
        return(real_log_contrast);
}
```